Naval Research Laboratory

Stennis Space Center, MS 39529-5004

NRL/MR/7320--12-9386

Global Ocean Forecast System V3.0 Validation Test Report Addendum: Provision of Boundary Conditions to the Relocatable Navy Coastal Ocean Model (NCOM))

E.J. METZGER
P.G. THOPPIL

Ocean Dynamics and Prediction Branch
Oceanography Division

G. PEGGION
University of New Orleans
New Orleans, Louisiana

D.S. FRANKLIN
O.M. SMEDSTAD

QinetiQ North America
Technology Solutions Group
Slidell, Louisiana

March 21, 2012

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
21-03-2012	Memorandum Report	

4. TITLE AND SUBTITLE

Global Ocean Forecast System V3.0 Validation Test Report Addendum: Provision of Boundary Conditions to the Relocatable Navy Coastal Ocean Model (NCOM)

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
0603207N

6. AUTHOR(S)

E.J. Metzger, P.G. Thoppil, G. Peggion,* D.S. Franklin,† and O.M. Smedstad†

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
73-5094-12-5

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
Oceanography Division
Stennis Space Center, MS 39529-5004

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/MR/7320--12-9386

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Space & Naval Warfare Systems Command
2451 Crystal Drive
Arlington, VA 22245-5200

10. SPONSOR / MONITOR'S ACRONYM(S)

SPAWAR

11. SPONSOR / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

*University of New Orleans, Work 2000 Lakeshore Drive, New Orleans, Louisiana 70148
†QinetiQ North America, Technology Solutions Group, Slidell, Louisianna

14. ABSTRACT

One of the primary functions of the Global Ocean Forecast System (GOFS) Version 3.0 (V3.0) (consisting of the 1/12° global HYbrid Coordinate Ocean Model (HYCOM) that employs the Navy Coupled Ocean Data Assimilation (NCODA)) is to provide boundary conditions (BCs) to higher horizontal and vertical resolution regional nested ocean models, such as the Relocatable (Relo) Navy Coastal Ocean Model (NCOM). This document can be viewed as both a User's Manual and an addendum to the GOFS V3.0 Phase II boundary condition validation work. As such, the dual methodology for extracting GOFS V3.0 BCs is documented. Secondly, it describes a set of experiments in which BCs extracted from the real-time GOFS V3.0 have been remapped to 40, 50, or 100 vertical levels, and these are used with corresponding versions of Relo NCOM configured for the region off the New Jersey coast. The sensitivity of the vertical resolution is examined with regard to error analyses of temperature (T) vs depth and acoustical proxy measures (i.e., mixed layer depth (MLD), sonic layer depth (SLD), below layer gradient (BLG), and deep sound channel (DSC)).

15. SUBJECT TERMS

Boundary conditions
Global HYCOM
Relocatable NCOM
GOFS V3.0

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			E. Joseph Metzger
Unclassified Unlimited	Unclassified Unlimited	Unclassified Unlimited	UU	54	19b. TELEPHONE NUMBER *(include area code)* (228) 688-4762

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

1.0 INTRODUCTION

One of the primary functions of the Global Ocean Forecast System (GOFS) Version 3.0 (V3.0) (consisting of the 1/12º global HYbrid Coordinate Ocean Model (HYCOM) that employs the Navy Coupled Ocean Data Assimilation (NCODA), Metzger et al., 2008) is to provide boundary conditions (BCs) to higher horizontal and vertical resolution regional nested ocean models. This was validated in the GOFS V3.0 Phase II Validation Test Report (VTR) (Metzger et al., 2010) for the Relocatable (Relo) (or Regional) Navy Coastal Ocean Model (NCOM) (Rowley et al., 2010) configured for the region surrounding Luzon Strait that connects the Pacific Ocean with the South China Sea. The Phase II VTR determined that the Luzon Strait Relo NCOM hindcast using GOFS V3.0 BCs produced superior results to a twin hindcast that used BCs from GOFS V2.6 (consisting of 1/8º global NCOM/NCODA, 1/32º Navy Layered Ocean Model (NLOM) and 1/8º Modular Ocean Data Assimilation (MODAS)).

This document can be viewed as both a User's Manual and an addendum to the Phase II boundary condition validation work. As such, the dual methodology for extracting GOFS V3.0 BCs is documented (Section 2). Secondly, it describes a set of experiments in which BCs extracted from the real-time GOFS V3.0 have been remapped to 40, 50 or 100 vertical levels and these are used with corresponding versions of Relo NCOM configured for the region off the New Jersey coast (Section 3). The sensitivity of the vertical resolution is examined with regard to error analyses of temperature (T) vs. depth and acoustical proxy measures (i.e. mixed layer depth (MLD), sonic layer depth (SLD), below layer gradient (BLG) and deep sound channel (DSC)).

Manuscript approved January 11, 2012.

2.0 METHODOLOGY FOR EXTRACTING GOFS V3.0 BCs

There are currently two sets of procedures for extracting GOFS V3.0 BCs for use with Relo NCOM. Under method 1, as part of the real-time system's runstream, netCDF files are output on a constant .08° latitude/longitude grid that spans the globe from 80°S-80°N. (Currently these netCDF files are output from the nowcast through the 120-hr forecast with 3-hourly temporal frequency.) These are vertically interpolated to 40 levels: 0, 2, 4, 6, 8, 10, 12, 15, 20, 25, 30, 35, 40, 45, 50, 60, 70, 80, 90, 100, 125, 150, 200, 250, 300, 350, 400, 500, 600, 700, 800, 900, 1000, 1250, 1500, 2000, 2500, 3000, 4000 and 5000 m. A sequence of scripts (defined in Section 2.1) is then executed to extract the BCs for a user-defined Relo NCOM domain. Under method 2, the global 3D HYbrid Coordinate Ocean Model (HYCOM) archive files on the native vertical hybrid grid (or on a user-defined subdomain) can be extracted and saved at a defined temporal frequency. A different sequence of scripts (defined in Section 2.2) is then executed to extract the BCs for a Relo NCOM domain. An advantage of the second method is that the vertical resolution can be different from the 40 levels defined in the first approach.

The scripts defined in this document have been saved using the Subversion (SVN) configuration management software and can be found at:

setenv SVNROOT "https://www7320.nrlssc.navy.mil/svn/repos"

${SVNROOT}/GOFSV3/trunk/postproc/BC_extraction/V1.0/script_cut

${SVNROOT}/GOFSV3/trunk/postproc/BC_extraction/V1.0/script_glb

${SVNROOT}/GOFSV3/trunk/postproc/BC_extraction/V1.0/subregion

2.1 Method 1: Procedure to extract BCs from GOFS V3.0 netCDF files

The basic idea behind both methods is to create netCDF files with the same naming convention and format that Relo NCOM uses for receiving BCs from GOFS V2.6. Thus it is necessary to create the files and directory structure similar to the hostnl namelist of the Relo NCOM domain, a portion of which is noted below:

```
&hostnl
   host_ncpath = '/scr/${user}/hycom/Nc_GLBu0.08
   host_odimens = '/scr/${user}/hycom/GLBu0.08/input0/odimens.D',
   host_ohgrda = '/scr/${user}/hycom/GLBu0.08/input0/ohgrd_1.A',
   host_ohgrdb = '/scr/${user}/hycom/GLBu0.08/input0/ohgrd_1.B',
   host_ovgrdd = '/scr/${user}/hycom/GLBu0.08/input0/ovgrd_1.D',
```

2.1.1 Script set-up

It is user's responsibility to create/copy/modify the files and to define the environmental variables. In the scripts that follow, the yellow highlighted areas may need to be changed by the user.

1. The file **/u/home/${user}/hycom_bin.env** contains the location of the executables and working directories required for extracting GOFS V3.0 BCs.

/u/home/${user}/hycom_bin.env

```
#
# --- This file defines environmental variables that are used
# --- in extracting GOFS V3.0 boundary conditions for a Relo
# --- NCOM nested domain
#
# --- working directory
#
setenv WRK_dir    /scr/${user}/hycom
#
# --- directory for scripts to extract BCs:
# --- SCRPT_cut is the methodology for using HYCOM archive files
# ---   that are a subregion of the entire domain
# --- SCRPT_glb is the methodology for using the constant .08 deg
# ---   lat/lon netCDF files
#
setenv SCRPT_cut  /u/home/${user}/hyc2ncom/script_cut
setenv SCRPT_glb  /u/home/${user}/hyc2ncom/script_glb
```

```
#
# --- netCDF operators directory
#
setenv NCO        /site/nco-3.9.8_64/bin/
#
# --- executables and utilities directories
#
setenv AWBIN      /u/home/wallcraf/hycom/ALL/bin
setenv AWBIN2     /u/home/wallcraf/hycom/ALL/ncom/src/
setenv AWsub      /u/home/wallcraf/hycom/ALL/subregion/src
setenv AWarc      /u/home/wallcraf/hycom/ALL/archive/src
setenv AWTOPO     /u/home/wallcraf/hycom/ALL/topo/src
setenv MODASbin   /u/home/smedstad/bin
setenv PLTF       navo
#
# --- topography directories: note GLBa0.08 is the native Mercator-
# --- curvilinear HYCOM grid whereas GLBu0.08 is the constant .08
# --- deg lat/lon grid onto which the netcdf files have been
# --- interpolated
#
setenv TOPOhy_dir /u/home/${user}/hycom/GLBa0.08/topo
setenv TOPOnc_dir newton:/u/home/${user}/hycom/GLBu0.08/topo
```

2. The scripts in the directory /u/home/${user}/hyc2ncom/script_glb are used on the GOFS V3.0 constant .08° lat/lon netCDF files, i.e. the GLBu0.08 grid. The following files are required:

a. **/u/home/${user}/hyc2ncom/script_glb/include.env** defines more environmental variables, the time frame over which to extract BCs, the GLBu0.08 array size and the GOFS V3.0 experiment number, in this example 908. Typically, day1, day2 and maybe taus will be the only variables to change in this file.

```
#!/bin/csh
#
# --- this file is used as part of the methodology to extract
# --- GOFS V3.0 BCs for a Relo NCOM domain. In this case, GOFS
# --- V3.0 input files are in netCDF format and on a constant
# --- 0.08 deg lat/lon grid
#
source /u/home/${user}/hycom_bin.env
setenv SCRPT     $SCRPT_glb
#
setenv AREA      GLBu0.08
setenv ncom_gr   $WRK_dir/$AREA/input0
setenv ARCHenv   $WRK_dir/global_nc
setenv NC_area   $WRK_dir/Nc_$AREA
setenv Topo_area $ARCHenv
```

```
#
# --- day1 = start DTG, day2 = end DTG, taus = temporal freq
#
setenv day1 20100820
setenv day2 20100821
setenv taus "000 003 006 009 012 015 018 021"
#
# --- array dimensions of GOFS V3.0 netcdf files
#
setenv NX 4500
setenv NY 2001
#
# --- more GOFS V3.0 parameters:
# --- Tenv   = topography version number
# --- REGenv = domain name for the constant .08 deg grid
#
setenv Tenv 09
setenv REGenv GLBu0.08
#
# --- expt name in two forms
#
setenv Eenv 908
setenv Xenv 90.8
#
# --- HRenv  = analysis time
# --- HR2env = nowcast time
#
setenv HRenv 18
setenv HR2env 00
```

b. **/u/home/${user}/hyc2ncom/script_glb/step1.get_GOFSV3_ncdf.com** is used to transfer the GLBu0.08 netCDF files from the archive machine (newton.navo.hpc.mil) to the working directory.

```
#!/bin/csh -x
#PBS -N XXX
#PBS -j oe
#PBS -o XXX.log
#PBS -e XXX.err
#PBS -l walltime=6:00:00
#PBS -A NRLSS018
#PBS -q transfer
#PBS -l select=1
#
set echo
set time = 1
#
cd
```

```
source /u/home/${user}/hycom_bin.env
source $SCRPT_glb/include.env
#
# --- location of GOFS V3.0 netCDF files interpolated to constant
# --- 0.08 deg lat/lon grid
#
setenv V /u/home/ooc/data/hycom/${REGenv}/expt_${Xenv}/data/netcdf
setenv S $ARCHenv
#
if (! -e ${S}) mkdir -p ${S}
cd ${S}
set var = "ssh ts3z uv3z"
set tau = "000 003 006 009 012 015 018 021"
#
set dstr = $day1
set dend = $day2
echo $dstr $dend
#
# --- stage files on newton
#
while ( $dstr <= $dend)
 foreach nm ($var)
  foreach tt ($tau)
   rsh newton /opt/SUNWsamfs/bin/stage
${V}/hycom_glb_${Eenv}_${dstr}00_t${tt}_${nm}.nc
  end
 end
 set dstr =  `$MODASbin/addndays YYYYMMDD $dstr +1`
end
#
set dstr = $day1
set dend = $day2
#
# --- copy files from newton
#
while  ( $dstr <= $dend )
 foreach nm ($var)
  foreach tt ($tau)
   /usr/bin/rcp
newton:${V}/hycom_glb_${Eenv}_${dstr}00_t${tt}_${nm}.nc . &
  end
 end
 wait
 set dstr =  `$MODASbin/addndays YYYYMMDD $dstr +1`
end
```

c. **/u/home/${user}/hyc2ncom/script_glb/step2.manipulate_ncdf.com** is used to manipulate the GLBu0.08 netCDF files and put in a format consistent with what Relo NCOM expects.

```
#!/bin/csh -f
#PBS -N XXX
#PBS -j oe
#PBS -o XXX.log
#PBS -e XXX.err
#PBS -l walltime=3:00:00
#PBS -A NRLSS018
#PBS -q standard
#PBS -l select=1
#
set echo
set time = 1
#
# --- this script manipulates the netcdf files and modifies to
# --- make them consistent with what Relo NCOM wants
#
/bin/date
cd
source /u/home/${user}/hycom_bin.env
source $SCRPT_glb/include.env
#
setenv Nncks $NCO/ncks
setenv WK    $NC_area
mkdir -p     $WK
#
setenv ARCH $ARCHenv
#
set taus = '000 003 006 009 012 015 018 021'
set dstr = $day1
set dend = $day2
#
set var = "ssh s3d t3d u3d v3d"
set hnm = hycom_$Eenv
#
while($dstr <= $dend)
 foreach dd ($taus)
  set glfile = $ARCH/hycom_glb_$Eenv"_"$dstr"00_t"$dd

  /bin/cp $glfile"_"ssh.nc $WK/ssh_$hnm"_"$dstr"00_t"$dd"h".nc
  $Nncks -O -h -v water_temp $glfile"_"ts3z.nc
$WK/t3d_$hnm"_"$dstr"00_t"$dd"h".nc
  $Nncks -O -h -v salinity $glfile"_"ts3z.nc
$WK/s3d_$hnm"_"$dstr"00_t"$dd"h".nc
  $Nncks -O -h -v water_u $glfile"_"uv3z.nc
$WK/u3d_$hnm"_"$dstr"00_t"$dd"h".nc
  $Nncks -O -h -v water_v $glfile"_"uv3z.nc
$WK/v3d_$hnm"_"$dstr"00_t"$dd"h".nc

  foreach name ($var)
   set file = $WK/$name"_"$hnm"_"$dstr"00_t"$dd"h".nc
   $NCO/ncwa -O -h -a time $file $file
  end

  echo ncwa $dd
  set file = $hnm"_"$dstr"00_t"$dd"h".nc
  $Nncks -O -h -v surf_el $WK/ssh_$file $WK/ssh_$file
  $Nncks -O -h -v water_temp $WK/t3d_$file $WK/t3d_$file
  $Nncks -O -h -v salinity $WK/s3d_$file $WK/s3d_$file
```

```
  $Nncks -O -h -v water_u $WK/u3d_$file $WK/u3d_$file
  $Nncks -O -h -v water_v $WK/v3d_$file $WK/v3d_$file
 end
 echo $dstr done
 set dstr = `$MODASbin/addndays YYYYMMDD $dstr +1`
end
#
/bin/date
```

d. **/u/home/${user}/hyc2ncom/script_glb/get_ncgrid.com** is used to transfer from newton.navo.hpc.mil the files associated with the GOFS V3.0 constant .08 deg lat/lon grid.

```
#!/bin/csh
#PBS -N XXX
#PBS -j oe
#PBS -o XXX.log
#PBS -e XXX.err
#PBS -l walltime=1:00:00
#PBS -A NRLSS018
#PBS -q transfer
#PBS -l select=1
#
# --- script to transfer from newton the files associated
# --- with the GOFS V3.0 constant .08 deg lat/lon grid
#
set echo
set time = 1
#
cd
source /u/home/${user}/hycom_bin.env
source $SCRPT_glb/include.env
#
setenv V /u/home/${user}/hycom/$AREA/input0
setenv S $ncom_gr
#
if (! -e ${S}) mkdir -p ${S}
cd ${S}
#
# --- stage files on newton
#
set file = "odimens.D ohgrd_1.A ohgrd_1.B ovgrd_1.D"
foreach ff ($file)
 /usr/bin/rsh newton /opt/SUNWsamfs/bin/stage ${V}/$ff
end
#
# --- copy files from newton
#
foreach ff ($file)
  /usr/bin/rcp newton:${V}/$ff . &
end
wait
```

2.1.2 Executing the scripts to extract GOFS V3.0 BCs

After the environmental variables are set, the scripts are run in this order.

1. **step1.get_GOFSV3_ncdf.com:** to retrieve the GOFS V3.0 netCDF files and store them in $ARCHenv

2. **step2.manipulate_ncdf.com:** to convert the GOFS V3.0 netCDF files in $ARCHenv and store the new format in $NC_area.

3. **step3.get_GLBu0.08.com:** to retrieve the $ncom_gr, the input0 directory of hostnl

4. Specify in **hostnl**, the path on $ncom_gr and $ NC_area.

5. Run Relo NCOM.

2.2 Method 2: Procedure to extract BCs from the GOFS V3.0 archive files on the native hybrid vertical grid

Boundary conditions can also be extracted from GOFS V3.0 archive files on the native hybrid vertical grid. This procedure is more cumbersome but has the advantage of allowing the vertical resolution to be different from the 40 pre-defined levels used in the GOFS V3.0 netCDF files. The input GOFS V3.0 archive files can cover the whole domain, or just a subdomain (defined in the archt.input file in the working GOFS V3.0 directory). If whole domain files are used, a sequence of scripts is used to extract a portion of the entire domain that is slightly larger than the Relo NCOM domain. These smaller files are then easier to manage. The procedure to process the GOFS V3.0 *subdomain* archive files is detailed in the appendix and this must be done before the following steps are performed. Such subdomain files would have to be created as GOFS V3.0 is running.

The example that follows is set up for a Relo NCOM domain that covers the Luzon Strait region. The longitude and latitude ranges are: 118.825-126.274°E and 17.35-25.153°N and it is

configured with 50 vertical levels. The GOFS V3.0 archive files must be slightly larger than the Relo NCOM domain and this example will use 118.0-127.0°E, 17.0-26.0°N. The user must determine the number of GOFS V3.0 gridpoints in the west-east (NX) and north-south (NY) directions and this is done as follows:

```
d13n6 88> /u/home/wallcraf/hycom/ALL/bin/hycom_lonlat2ij 118.0 17.0
~/hycom/GLBa0.08/topo/regional.grid.a
  549 1721
d13n6 89> /u/home/wallcraf/hycom/ALL/bin/hycom_lonlat2ij 127.0 26.0
~/hycom/GLBa0.08/topo/regional.grid.a
  661 1842
```

So, NX = 661 – 549 + 1 = 113 and NY = 1842 -1721 + 1 = 122.

1. As in the previous methodology, the file **/u/home/${user}/hycom_bin.env** contains the location of the executables and working directories required for extracting GOFS V3.0 BCs.

2. The scripts in the directory **/u/home/${user}/hyc2ncom/script_cut** are used on the GOFS V3.0 archive files on the native hybrid vertical grid. The following files are required:

a. **/u/home/${user}/hyc2ncom/script_cut/include.env** defines more environmental variables, the time frame over which to extract BCs, the array size and the experiment number, in this example 908. Typically the only variables that change in this file are: day1/day2 – the start/end dates and NX/NY – the longitude/latitude dimensions on the GLBa0.08 grid (which have been determined above).

```
#!/bin/csh
#
# --- this file is used as part of the methodology to extract
# --- GOFS V3.0 BCs for a Relo NCOM domain. In this case, GOFS
# --- V3.0 input files are a subdomain of the full model grid
# --- and on the hybrid vertical grid, i.e. not z-levels
#
source /u/home/${user}/hycom_bin.env
setenv SCRPT $SCRPT_cut
```

```
#
# --- array dimensions of the GOFS V3.0 subdomain
#
setenv NX  113
setenv NY  122
#
# --- environmental variable to define how the Relo NCOM vertical
# --- is configured. When set to 1, use pre-defined depths, when set
# --- to 0 use the Relo NCOM algorithm
#
setenv MAKE_VGRID 1
#
# --- day1 = start DTG, day2 = end DTG
#
setenv day1 20100702
setenv day2 20100706
setenv year `echo $day1 | awk '{print substr($1,1,4)}'`
#
# --- directories
#
setenv TMP       $WRK_dir/tmp
setenv AREA      LuzonStrait_cut
setenv ncom_gr   $WRK_dir/$AREA/input0
setenv Topo_area /u/home/${user}/hyc2ncom/Topo_$AREA
setenv ARCHenv   $WRK_dir/global
setenv TOPO_dir  $TOPOhy_dir
setenv ARCH_area $WRK_dir/Arch_$AREA
setenv NC_area   $WRK_dir/Nc_$AREA
#
if (! -e $Topo_area) mkdir -p $Topo_area
if (! -e $ARCH_area) mkdir -p $ARCH_area
if (! -e $NC_area)   mkdir -p $NC_area
#
# --- more GOFS V3.0 parameters:
# --- Tenv   = topography version number
# --- REGenv = domain name for GOFS V3.0
#
setenv Tenv 09
setenv REGenv GLBa0.08
#
# --- expt number
#
setenv Eenv 908
#
# --- HRenv  = analysis time
# --- HR2env = nowcast time
#
setenv HRenv 18
setenv HR2env 00
#
# --- define Julian dates
#
if ( $PLTF == navo) then
  set mm = `echo $day1 | awk '{print substr($1,5,2)}'`
  set dd = `echo $day1 | awk '{print substr($1,7,2)}'`
  set jj = `csh $MODASbin/ymd2doy.csh $year $mm $dd`
  setenv jul1 `echo $jj  | awk '{print substr($1,6,3)}'`
```

```
  set mm = `echo $day2 | awk '{print substr($1,5,2)}'`
  set dd = `echo $day2 | awk '{print substr($1,7,2)}'`
  csh $MODASbin/ymd2doy.csh $year $mm $dd
  set jj = `csh $MODASbin/ymd2doy.csh $year $mm $dd`
  setenv jul2 `echo $jj  | awk '{print substr($1,6,3)}'`
else
  setenv jul1 `$MODASbin/idtg2jul $day1`
  setenv jul2 `$MODASbin/idtg2jul $day2`
endif
```

b. **/u/home/${user}/hyc2ncom/script_cut/step1.get_GOFSV3_arch.com** will transfer from newton.navo.hpc.mil the tarballs that contains the best GOFS V3.0 analysis for each day of a specific month and year.

```
#!/bin/csh
#
#PBS -N XXX
#PBS -j oe
#PBS -o XXX.log
#PBS -e XXX.err
#PBS -l walltime=6:00:00
#PBS -A NRLSS018
#PBS -q transfer
#PBS -l select=1
#
set echo
set time = 1
#
# --- this script will transfer from newton the tarballs that contains
# --- the best GOFS V3.0 analysis for each day of a specific month and
# --- year
#
cd
source ~/hycom_bin.env
source $SCRPT_cut/include.env
#
setenv R GLBa0.08
setenv E 908
setenv X 90.8
setenv Y 2010
setenv M 07
setenv V /u/home/ooc/data/hycom/${R}/expt_${X}/data
setenv S /scr/${user}/hycom/global
#
if (! -e ${S}) then
  mkdir -p ${S}
endif
cd ${S}
#
# --- list all best analysis tarballs for a given month into a file;
# --- sort through and find the best analysis
#
# --- the first day of the month is special
#
```

```
rsh newton "cd ${V}; ls ${E}_archv_${Y}??????_${Y}${M}0100*gz" >! l1
rsh newton "cd ${V}; ls ${E}_archv_${Y}??????_${Y}${M}*gz"     >! l2
cat l1 l2 >! list1
#
set L = `wc -l list1 | awk '{printf("%d", $1-1)}'`
tail -n ${L} list1 >! list2
#
cat list2 | sed -e 's/_/ /g' -e 's/\./ /g' | sort -nr -k 4 | uniq -f 3
| sort -n -k 4 | sed -e 's/ gz/.gz/g' -e 's/ tar/.tar/g' -e 's/ /_/g'
>! list1
#
set L = `wc -l list1 | awk '{printf("%d", $1)}'`
#
cat list1
#
# --- copy files from newton
#
@ N = 1
while (${N} <= ${L})
  set F = `head -n ${N} list1 | tail -n 1`
  rcp newton:${V}/${F} . &
  @ N ++
end
wait
#
# --- untar/uncompress tarballs
#
@ N = 1
while (${N} <= ${L})
  set F = `head -n ${N} list1 | tail -n 1`
  /site/unsupported/bin/gtar --format=posix -xvzf ${F} &
  @ N ++
end
wait
#
# --- delete tarballs
#
@ N = 1
while (${N} <= ${L})
  set F = `head -n ${N} list1 | tail -n 1`
  /bin/rm -f ${F}
  @ N ++
end
#
/bin/rm -f list[12] l[12]
```

c. **/u/home/${user}/hyc2ncom/script_cut/step2.get_GLBa0.08.com** will link or transfer from newton.navo.hpc.mil the needed topography and latitude/longitude definition files.

```
#!/bin/csh
#
#PBS -N XXX
#PBS -j oe
#PBS -o XXX.log
#PBS -e XXX.err
```

```
#PBS -l walltime=6:00:00
#PBS -A NRLSS018
#PBS -q transfer
#PBS -l select=1
#
set echo
set time = 1
#
cd
source /u/home/${user}/hycom_bin.env
source $SCRPT_cut/include.env

setenv V /u/home/${user}/hycom/${REGenv}/topo
setenv S $ARCHenv
setenv LINK 1
#
if (! -e ${S}) mkdir -p ${S}
cd ${S}
#
if ($LINK == 1) then
 /bin/rm regional.*
 ln -sf ${V}/regional.grid.a .
 ln -sf ${V}/regional.grid.b .
 ln -sf ${V}/depth_${REGenv}_${Tenv}.b regional.depth.b
 ln -sf ${V}/depth_${REGenv}_${Tenv}.a regional.depth.a
else
 rsh newton /opt/SUNWsamfs/bin/stage ${V}/regional.grid.[ab]
 rcp newton:${V}/regional.grid.a . &
 rcp newton:${V}/regional.grid.b . &
 rsh newton /opt/SUNWsamfs/bin/stage ${V}/depth_${REGenv}_${Tenv}*
 rcp newton:${V}/depth_${REGenv}_${Tenv}.a regional.depth.a
 rcp newton:${V}/depth_${REGenv}_${Tenv}.b regional.depth.b
endif
wait
```

3. Next the Relo NCOM horizontal grid must be configured. It is assumed the user has sufficient knowledge to do this. Edit **/u/home/${user}/hyc2ncom/script_cut/relo.nl**. Here, variables `m` = $NX and `n` = $NY, `rlon`/`rlat` are the starting longitude/latitude, `dmax` is the deepest level and the first element of `lo` defines the number of levels. NOTE: STEPS 3-7 ONLY HAVE TO BE DONE ONE TIME.

```
&dsetnl
/
&gridnl
  nnest = 1,
  nproj = 5,
  m     = 113,  !longitude dimensions, m must be before n
  n     = 122,  !latitude  dimensions
  rlon  = 118.0 !starting longitude
  rlat  = 17.0  !starting latitude
```

```
  iref  = 1,
  jref  = 1,
  ii    = 1,
  jj    = 1,
  delx  = 0.0833333,
  dely  = 0.0833333,
/
&hostnl
/
&oanl
/
&omnloff
/
&parmlst
/
&rlx3nl
/
&setupl
       bathyfile = '/u/home/ooc/models/relo/relo_etc/dbdb2_v30.dat',
            dmax =    -5500.,
            dmin =      -10.,
           dztop =       .5,
        gdem_dir = '/u/home/rowley/usr/relo/etc//ncoda/gdem3s',
         gdemfile = '/u/home/rowley/usr/relo/etc/gdem3_ts0.dat',
     initialtide =   .false.,
              lo =       50,       2,       50,       50,       50,
                         50,       50,
             lso =        2,       35,       35,       35,       35,
                         35,       35,
         nobmaxo =     4000,     4000,     4000,     1000,     1000,
                       1000,     1000,
             nqo =        2,        2,        2,        2,        2,
                          2,        2,
           nrivo =      200,       50,       50,       50,       50,
                         50,       50,
             nro =        2,        2,        2,        2,        2,
                          2,        2,
            ntco =        8,        8,        8,        8,        8,
                          8,        8,
           ntypo =        1,        1,        1,        1,        1,
                          1,        1,
       riverfile =
'/net/dynamic/export/data/rowley/models/relo/relo_1.1/etc/rivers6.dat'
,
     startatrest =  .false.,
        tidefile =
'/net/dynamic/export/data/rowley/models/relo/relo_1.1/etc/tide_egb.dat
',
       writeinit =  .true.,
      writeosstf =  .false.,
       writeotsf =  .false.,
/
&sflxnl
/
```

a. There are two options for creating the Relo NCOM vertical grid (ovgrid) and these are controlled by the environmental variable MAKE_VGRID found in **include.env**. The scripts **/u/home/${user}/hyc2ncom/script_cut/step3.make_reloncom.*** define the procedure. If MAKE_VGRID = 1, (i.e. pre-defined depths), an ASCII file (zin.dat) with the appropriate number of depths and levels must exist in $SCRPT_cut/bin.

/u/home/${user}/hyc2ncom/script_cut/bin/zin.dat

```
   50      'number of levels'
    0.000  'z     ' = sample depth
    0.500  'z     ' = sample depth
    1.083  'z     ' = sample depth
    1.762  'z     ' = sample depth
    2.554  'z     ' = sample depth
    3.477  'z     ' = sample depth
    4.552  'z     ' = sample depth
    5.806  'z     ' = sample depth
    7.268  'z     ' = sample depth
    8.971  'z     ' = sample depth
   10.957  'z     ' = sample depth
   13.271  'z     ' = sample depth
   15.968  'z     ' = sample depth
   19.112  'z     ' = sample depth
   22.777  'z     ' = sample depth
   27.049  'z     ' = sample depth
   32.027  'z     ' = sample depth
   37.831  'z     ' = sample depth
   44.595  'z     ' = sample depth
   52.479  'z     ' = sample depth
   61.669  'z     ' = sample depth
   72.380  'z     ' = sample depth
   84.865  'z     ' = sample depth
   99.418  'z     ' = sample depth
  116.380  'z     ' = sample depth
  136.151  'z     ' = sample depth
  159.195  'z     ' = sample depth
  186.055  'z     ' = sample depth
  217.363  'z     ' = sample depth
  253.855  'z     ' = sample depth
  296.390  'z     ' = sample depth
  345.968  'z     ' = sample depth
  403.755  'z     ' = sample depth
  471.110  'z     ' = sample depth
  549.619  'z     ' = sample depth
  641.128  'z     ' = sample depth
  747.789  'z     ' = sample depth
  872.111  'z     ' = sample depth
 1017.019  'z     ' = sample depth
 1185.922  'z     ' = sample depth
 1382.793  'z     ' = sample depth
```

```
  1612.263  'z     ' = sample depth
  1879.729  'z     ' = sample depth
  2191.483  'z     ' = sample depth
  2554.860  'z     ' = sample depth
  2978.406  'z     ' = sample depth
  3472.000  'z     ' = sample depth
  4047.000  'z     ' = sample depth
  4718.000  'z     ' = sample depth
  5500.000  'z     ' = sample depth
```

If MAKE_VGRID = 0, the Relo NCOM algorithm will be used to define the depths and levels.

/u/home/${user}/hyc2ncom/script_cut/step3.make_reloncom.com

```
#!/bin/csh

source /u/home/${user}/hycom_bin.env
source $SCRPT_cut/include.env

$SCRPT/step3.make_reloncom.s
```

/u/home/${user}/hyc2ncom/script_cut/step3.make_reloncom.s

```
#!/bin/sh

export RELO=/u/home/rowley/usr/relo
export BINDIR=$RELO/bin
export JOBDIR=$RELO/script
export ETCDIR=$RELO/etc
export DEFDIR=$RELO/default
export STATIC=$WRK_dir
export REGION=$AREA
mkdir -p $STATIC/$REGION
mkdir -p $STATIC/$REGION/input0

cd $STATIC/$REGION

export NCOM_OHGRD_1B=input0/ohgrd_1.B
export NCOM_OHGRD_1A=input0/ohgrd_1.A
export NCOM_DIMEN_0D=input0/odimens.D
export NCOM_OVGRD_1D=input0/ovgrd_1.D
export NCOM_OZOUT_1D=input0/ozout_1.D

#ovgrid from relonl
if [ $MAKE_VGRID -eq 0 ]; then
$BINDIR/ncom_config.xc -dovgrd relo.nl >input0/ovgrid.log
else
csh $SCRPT/make_ovgrid.com
fi

# make the ohgrid for Relo NCOM
/bin/cp $SCRPT/relo.nl $STATIC/$REGION
$BINDIR/ncom_config.xc -dohgrd relo.nl

cd $SCRPT
```

/u/home/${user}/hyc2ncom/script_cut/make_ovgrid.com

```
#!/bin/csh
#
source /u/home/${user}/hycom_bin.env
source $SCRPT_cut/include.env
#
cd $SCRPT/bin
#
/bin/rm odimens.D
/bin/rm ovgrd_1.D
ln -s $ncom_gr/odimens.D odimens.D
ln -s $ncom_gr/ovgrd_1.D ovgrd_1.D
#
$SCRPT/bin/z2ovgrid.exe <<E-o-D
 &inputs
   Nx=$NX
   Ny=$NY
 /
E-o-D
cd $SCRPT
```

b. **/u/home/${user}/hyc2ncom/script_cut/step4.make_regiongrid.com** creates a HYCOM version of the Relo NCOM grid based on the NCOM longitude and latitude fields.

```
#!/bin/csh
#
set echo
#
cd
source /u/home/${user}/hycom_bin.env
source $SCRPT_cut/include.env
#
# --- a HYCOM version of a NCOM grid.
# --- based on the NCOM longitude and latitude fields.
#
setenv N   $ncom_gr
setenv IDM $NX
setenv JDM $NY
#
cd $Topo_area
/bin/rm -f regional.lonlat.a+ regional.lonlat.a
#
$AWBIN/raw2hycom     $N/ohgrd_1.A       ${IDM} ${JDM}
regional.lonlat.a+
$AWBIN/hycom_extract regional.lonlat.a+ ${IDM} ${JDM} 2 1 1 1
regional.lonlat.a
/bin/rm -f    regional.lonlat.a+
#
# --- temporary regional.grid.b
#
cat >! regional.grid.b <<E-o-D
${IDM} 'idm   ' = longitudinal array size
${JDM} 'jdm   ' = latitudinal  array size
E-o-D
setenv FOR051A regional.lonlat.a
```

```
setenv FOR061A fort.61A
$AWTOPO/grid_lonlat_2d <<E-o-D
${IDM} 'idm   ' = longitudinal array size
${JDM} 'jdm   ' = latitudinal  array size
E-o-D
mv fort.61  regional.grid.b
mv fort.61A regional.grid.a

cd $SCRPT
```

c. **/u/home/${user}/hyc2ncom/script_cut/step5.make_gmapi.com** prepares the weight function of the interpolation from 2D native grid to 2D regular grid.

```
#!bin/csh -f
set echo
set time=1
#
cd
source /u/home/${user}/hycom_bin.env
source $SCRPT_cut/include.env
#
# --- form subregion grid array index map file, GLBa0.08 to the
# --- Luzon Strait cutout domain
#
setenv R $Topo_area
file ${R}/regional.grid.a
#
mkdir -p $TMP
cd $TMP
if( !(-e regional.grid.a)) then
  /bin/cp $TOPO_dir/regional.grid.a .
  /bin/cp $TOPO_dir/regional.grid.b .
endif
if( !(-e regional.depth.a)) then
  /bin/cp $TOPO_dir/regional.depth.a .
  /bin/cp $TOPO_dir/regional.depth.b .
endif
touch   ${R}/regional.gmapi_$REGenv.a
/bin/rm ${R}/regional.gmapi_$REGenv.[ab]
#
$AWsub/isuba_gmapi <<E-o-D
${R}/regional.grid.a
${R}/regional.gmapi_$REGenv.a
$REGenv (4500x3298) to $AREA (maxinc=25)
 $NX     'idm   ' = longitudinal array size of subregion
 $NY     'jdm   ' = latitudinal  array size
   25     'maxinc' = maximum input array index jump on target grid
E-o-D
#
cd $WRK_dir
```

d. **/u/home/${user}/hyc2ncom/script_cut/step6.depth_GLBu0.08.com** forms a subregion bathymetry file, GLBa0.08 to GLBu0.08.

```
#!/bin/csh
#
set echo
#
source /u/home/${user}/hycom_bin.env
source $SCRPT_cut/include.env
#
# --- form subregion bathymetry file, GLBa0.08 to Luzon Strait cutout
domain
#
setenv R $Topo_area
#
mkdir -p $TMP
cd $TMP
#
if( !(-e regional.grid.a)) then
  /bin/cp $TOPO_dir/regional.grid.a .
  /bin/cp $TOPO_dir/regional.grid.b .
endif
if( !(-e regional.depth.a)) then
  /bin/cp $TOPO_dir/regional.depth.a .
  /bin/cp $TOPO_dir/regional.depth.b .
endif
#
touch   ${R}/depth_${AREA}_${Tenv}.[ab]
/bin/rm ${R}/depth_${AREA}_${Tenv}.[ab]
#
$AWsub/isuba_topog <<E-o-D
${R}/regional.gmapi_GLBa0.08.a
$TOPO_dir/depth_${REGenv}_${Tenv}.b
${R}/depth_${AREA}_${Tenv}.b
depth_GLBa0.08 subregioned to GLBu0.08 via isuba_topog
 $NX      'idm   ' = longitudinal array size of subregion
 $NY      'jdm   ' = latitudinal  array size
E-o-D
```

e. **/u/home/${user}/hyc2ncom/script_cut/step7.reg_2_ncomgr.com** remakes the ohgrd Relo NCOM file with the GOFS V3.0 bathymetry.

```
set echo
#
# --- form NCOM ohgrd file for GLBa0.08
#
source /u/home/${user}/hycom_bin.env
source $SCRPT_cut/include.env
setenv D $Topo_area
setenv R $ncom_gr
#
/bin/rm  regional.grid.*
ln -sf $D/regional.grid.a .
ln -sf $D/regional.grid.b .
#
/bin/rm  regional.depth.*
ln -s $D/depth_$AREA"_"$Tenv.a regional.depth.a
ln -s $D/depth_$AREA"_"$Tenv.b regional.depth.b
#
```

```
setenv NCOM_DIMEN_0D $R/odimens.D
setenv NCOM_OVGRD_1D $R/ovgrd_1.D
#
setenv NCOM_OHGRD_1A $R/ohgrd_1.A
setenv NCOM_OHGRD_1B $R/ohgrd_1.B
#
touch   $NCOM_OHGRD_1A $NCOM_OHGRD_1B
touch   regional.dncom.b
/bin/rm regional.dncom.[ab]
/bin/mv $R/ohgrd_1.A $R/ohgrd_1.A.relo
/bin/mv $R/ohgrd_1.B $R/ohgrd_1.B.relo
#
$AWBIN2/grid2ncom <<E-o-D
  1      'i1st  ' = 1st hycom i-point on ncom grid
  1      'j1st  ' = 1st hycom j-point on ncom grid
E-o-D
#
/bin/rm regional.grid.[ab] regional.depth.[ab]
```

4. Convert the GOFS V3.0 archive files to Relo NCOM netCDF files. These scripts are computer intensive.

a. **/u/home/${user}/hyc2ncom/script_cut/step8.arch_2_ncdf.com** forms interpolated subregion archive files, GLBa0.08 to GLBu0.08.

```
#!/bin/csh
#PBS -N XXX
#PBS -j oe
#PBS -o XXX.log
#PBS -e XXX.log
#PBS -l walltime=2:00:00
#PBS -A NRLSS018
#PBS -q standard
#PBS -l select=1:ncpus=1
#
#
set echo
set time=1
#
set OS=`uname`
switch ($OS)
case 'Linux':
    which aprun
    if (! $status) then
      set APRUN='aprun -n 1 -m 15g '
      set SRC=/u/home/wallcraf/hycom/ALLcnl
    else
      set APRUN=''
      set SRC=/u/home/wallcraf/hycom/ALL
    endif
 breaksw
case 'AIX'
```

```
 set APRUN=''
 set SRC=/u/home/wallcraf/hycom/ALL
 breaksw
default:
 set APRUN=''
 set SRC=/u/home/wallcraf/hycom/ALL
endsw
#
date
#
cd
source /u/home/${user}/hycom_bin.env
source $SCRPT_cut/include.env
#
# --- form interpolated subregion archive files, GLBa0.08 to
# --- Luzon Strait cutout domain
#
# --- R is the original region
# --- U is the target   region
# --- D is the location of the original  archive files.
# --- N is the location of the subregion archive files.
# --- E,y,d,h select the archive files.
# --- T is topography number.
#
setenv T $Tenv
set REG=$REGenv
setenv U $AREA
set E=$Eenv
setenv X `echo ${E} | awk '{printf("%04.1f", $1*0.1)}'`
set EXPT=expt_${X}
#
setenv idtg $day1
setenv idtgtod $day2
# --- analysis time
setenv HR $HRenv
# --- nowcast time
setenv HR2 $HR2env
#
setenv INP $ARCHenv
setenv OUT $ARCH_area
mkdir -p         ${OUT}
lfs setstripe -d ${OUT}
lfs setstripe    ${OUT} 1048576 -1 8
#
cd ${INP}
#
if( !(-e ${INP}/regional.grid.a)) then
  /bin/cp  $TOPO_dir/regional.grid.a          ${INP}/regional.grid.a
  /bin/cp  $TOPO_dir/regional.grid.b          ${INP}/regional.grid.b
endif
if( !(-e ${INP}/regional.depth.a)) then
  /bin/cp $TOPO_dir/depth_${REG}_${T}.a       ${INP}/regional.depth.a
  /bin/cp $TOPO_dir/depth_${REG}_${T}.b       ${INP}/regional.depth.b
endif
#
if( !(-e ${OUT}/regional.gmap.a)) then
  /bin/cp  $Topo_area/regional.gmapi_${REG}.a ${OUT}/regional.gmap.a
  /bin/cp  $Topo_area/regional.gmapi_${REG}.b ${OUT}/regional.gmap.b
```

```
endif
if( !(-e ${OUT}/regional.grid.a)) then
  /bin/cp  $Topo_area/regional.grid.a         ${OUT}/regional.grid.a
  /bin/cp  $Topo_area/regional.grid.b         ${OUT}/regional.grid.b
endif
if( !(-e ${OUT}/regional.depth.a)) then
  /bin/cp  $Topo_area/depth_${U}_${T}.a       ${OUT}/regional.depth.a
  /bin/cp  $Topo_area/depth_${U}_${T}.b       ${OUT}/regional.depth.b
endif
#
set jul=$jul1
@ jul = $jul - 1
while ($jul <= $jul2)
#
  if (-e ${OUT}/archv_$AREA.${year}_${jul}_${HR2}.a) then
    /bin/rm ${OUT}/archv_$AREA.${year}_${jul}_${HR2}.*
  endif

  ${APRUN} ${AWsub}/isubaregion <<E-o-D
${OUT}/regional.grid.a
${OUT}/regional.gmap.a
${OUT}/regional.depth.a
${INP}/regional.depth.a
${INP}/${E}_archv.${year}_${jul}_${HR2}.a
${OUT}/archv_$AREA.${year}_${jul}_${HR2}.a
${REG} interpolated to ${U}
 $NX      'idm   ' = target longitudinal array size
 $NY      'jdm   ' = target latitudinal  array size
   1      'iceflg' = ice in output archive flag (0=none,1=energy loan
model)
   0      'smooth' = smooth interface depths    (0=F,1=T)
E-o-D

  @ jul = $jul + 1
end
#
date
```

b. **/u/home/${user}/hyc2ncom/script_cut/step9.ncdf_vertinterp.com.*** remaps the archive files to the defined vertical levels and creates netCDF files for Relo NCOM.

step9.ncdf_vertinterp.com

```
#!/bin/csh

#PBS -N XXX
#PBS -j oe
#PBS -e XXX.err
#PBS -o XXX.log
#PBS -l walltime=0:20:00
#PBS -A NRLSS018
#PBS -q standard
#PBS -l select=1:ncpus=1
#
set echo
set time = 1
```

```
#
cd
source /u/home/${user}/hycom_bin.env
source $SCRPT_cut/include.env
#
$SCRPT/step9.ncdf_vertinterp.s
```

step9.ncdf_vertinterp.s

```
#!/bin/sh
#
set echo
time = 1
#
# --- Script to "cutout" a hycom archive file to a netcdf file
# --- first do the 2d files and then the 3D files
# --- 3D files are extracted from hybrid into Z-coordinates
#
ncdf2d () {
#
# --- extract 2D fields from a single HYCOM archive file
#
CDF023=${NCDIR}/ssh"_"${DOMAIN}"_"${E}"_"${dtg}${tau}.nc
mkdir -p $NCDIR
#
touch   $CDF023
/bin/rm $CDF023
#
export CDF_TITLE="HYCOM $DOMAIN $EXPT"
export CDF_INST="Naval Research Laboratory"
export CDF023=$CDF023
$AWarc/archv2ncdf2d << EOF
${HYCOMINPUTFILE}
MERSEA
000     'iexpt ' = experiment number x10 (000=from archive file)
  3     'yrflag' = days in year flag (0=360J16,1=366J16,2=366J01,3-
actual)
 $NX    'idm   ' = longitudinal array size
 $NY    'jdm   ' = latitudinal  array size
  1     'kdm   ' = number of layers
 34.0   'thbase' = reference density (sigma units)
  0     'smooth' = smooth fields before plotting (0=F,1=T)
  0     'mthin ' = mask thin layers from plots   (0=F,1=T)
  1     'iorign' = i-origin of plotted subregion
  1     'jorign' = j-origin of plotted subregion
  0     'idmp  ' = i-extent of plotted subregion (<=idm; 0 implies
idm)
  0     'jdmp  ' = j-extent of plotted subregion (<=jdm; 0 implies
jdm)
  0     'botio ' = bathymetry       I/O unit (0 no I/O)
  0     'flxio ' = surf. heat flux  I/O unit (0 no I/O)
  0     'empio ' = surf. evap-pcip  I/O unit (0 no I/O)
  0     'ttrio ' = surf. temp trend I/O unit (0 no I/O)
  0     'strio ' = surf. saln trend I/O unit (0 no I/O)
  0     'icvio ' = ice coverage     I/O unit (0 no I/O)
  0     'ithio ' = ice thickness    I/O unit (0 no I/O)
  0     'ictio ' = ice temperature  I/O unit (0 no I/O)
 23     'sshio ' = sea surf. height I/O unit (0 no I/O)
```

```
  0     'bsfio ' = baro. strmfn.    I/O unit (0 no I/O)
  0     'uvmio ' = mix. lay. u-vel. I/O unit (0 no I/O)
  0     'vvmio ' = mix. lay. v-vel. I/O unit (0 no I/O)
  0     'spmio ' = mix. lay. speed  I/O unit (0 no I/O)
  0     'bltio ' = bnd. lay. thick. I/O unit (0 no I/O)
  0     'mltio ' = mix. lay. thick. I/O unit (0 no I/O)
  0     'sstio ' = mix. lay. temp.  I/O unit (0 no I/O)
  0     'sssio ' = mix. lay. saln.  I/O unit (0 no I/O)
  0     'ssdio ' = mix. lay. dens.  I/O unit (0 no I/O)
  1     'kf    ' = first output layer (=0 end output; <0 label with
layer #)
  1     'kl    ' = last  output layer
  0     'uvlio ' = layer k   u-vel. I/O unit (0 no I/O)
  0     'vvlio ' = layer k   v-vel. I/O unit (0 no I/O)
  0     'splio ' = layer k   speed. I/O unit (0 no I/O)
  0     'infio ' = layer k   i.dep. I/O unit (0 no I/O)
  0     'thkio ' = layer k   thick. I/O unit (0 no I/O)
  0     'temio ' = layer k   temp   I/O unit (0 no I/O)
  0     'salio ' = layer k   saln.  I/O unit (0 no I/O)
  0     'tthio ' = layer k   dens,  I/O unit (0 no I/O)
  0     'sfnio ' = layer k  strmfn. I/O unit (0 no I/O)
  0     'kf    ' = first output layer (=0 end output; <0 label with
layer #)
EOF

}
#
# --- extract 3D fields from a single HYCOM archive file
#
ncdf3d () {
#
# --- interpolate to 3D z-levels from a single HYCOM archive file
# --- and output to netCDF. Uses linear interpolation.
#
CDF033=${NCDIR}/u3d"_"${DOMAIN}"_"${E}"_"${dtg}${tau}.nc
CDF034=${NCDIR}/v3d"_"${DOMAIN}"_"${E}"_"${dtg}${tau}.nc
CDF035=${NCDIR}/t3d"_"${DOMAIN}"_"${E}"_"${dtg}${tau}.nc
CDF036=${NCDIR}/s3d"_"${DOMAIN}"_"${E}"_"${dtg}${tau}.nc
#
touch   $CDF033 $CDF034 $CDF035 $CDF036
/bin/rm $CDF033 $CDF034 $CDF035 $CDF036
#
export CDF_TITLE="HYCOM $DOMAIN $EXPT"
export CDF_INST="Naval Research Laboratory"
export CDF033=$CDF033
export CDF034=$CDF034
export CDF035=$CDF035
export CDF036=$CDF036
$AWarc/archv2ncdf3z << EOF
${HYCOMINPUTFILE}
MERSEA
 000    'iexpt ' = experiment number x10 (000=from archive file)
   3    'yrflag' = days in year flag (0=360J16,1=366J16,2=366J01,3-
actual)
 $NX    'idm   ' = longitudinal array size
 $NY    'jdm   ' = latitudinal  array size
  32    'kdm   ' = number of layers
  34.0  'thbase' = reference density (sigma units)
```

```
   0    'smooth' = smooth the layered fields (0=F,1=T)
   1    'iorign' = i-origin of plotted subregion
   1    'jorign' = j-origin of plotted subregion
   0    'idmp  ' = i-extent of plotted subregion (<=idm; 0 implies
idm)
   0    'jdmp  ' = j-extent of plotted subregion (<=jdm; 0 implies
jdm)
   1    'itype ' = interpolation type (0=sample,1=linear)
   50   'kz    ' = number of levels
     0.000  'z     ' = sample depth
     0.500  'z     ' = sample depth
     1.083  'z     ' = sample depth
     1.762  'z     ' = sample depth
     2.554  'z     ' = sample depth
     3.477  'z     ' = sample depth
     4.552  'z     ' = sample depth
     5.806  'z     ' = sample depth
     7.268  'z     ' = sample depth
     8.971  'z     ' = sample depth
    10.957  'z     ' = sample depth
    13.271  'z     ' = sample depth
    15.968  'z     ' = sample depth
    19.112  'z     ' = sample depth
    22.777  'z     ' = sample depth
    27.049  'z     ' = sample depth
    32.027  'z     ' = sample depth
    37.831  'z     ' = sample depth
    44.595  'z     ' = sample depth
    52.479  'z     ' = sample depth
    61.669  'z     ' = sample depth
    72.380  'z     ' = sample depth
    84.865  'z     ' = sample depth
    99.418  'z     ' = sample depth
   116.380  'z     ' = sample depth
   136.151  'z     ' = sample depth
   159.195  'z     ' = sample depth
   186.055  'z     ' = sample depth
   217.363  'z     ' = sample depth
   253.855  'z     ' = sample depth
   296.390  'z     ' = sample depth
   345.968  'z     ' = sample depth
   403.755  'z     ' = sample depth
   471.110  'z     ' = sample depth
   549.619  'z     ' = sample depth
   641.128  'z     ' = sample depth
   747.789  'z     ' = sample depth
   872.111  'z     ' = sample depth
  1017.019  'z     ' = sample depth
  1185.922  'z     ' = sample depth
  1382.793  'z     ' = sample depth
  1612.263  'z     ' = sample depth
  1879.729  'z     ' = sample depth
  2191.483  'z     ' = sample depth
  2554.860  'z     ' = sample depth
  2978.406  'z     ' = sample depth
  3472.000  'z     ' = sample depth
  4047.000  'z     ' = sample depth
  4718.000  'z     ' = sample depth
```

```
  5500.000  'z       ' = sample depth
    0    'botio ' = bathymetry  I/O unit (0 no I/O)
    0    'mltio ' = mix.l.thk.  I/O unit (0 no I/O)
    0    'tempml' = temperature jump across mixed-layer (degC,  0 no
I/O)
    0    'densml' =     density jump across mixed-layer (kg/m3, 0 no
I/O)
    0    'infio'  = interface depths  I/O unit (0 no I/O)
    0    'wvlio ' = w-velocity  I/O unit (0 no I/O)
   33    'uvlio ' = u-velocity  I/O unit (0 no I/O)
   34    'vvlio ' = v-velocity  I/O unit (0 no I/O)
    0    'splio ' = speed       I/O unit (0 no I/O)
   35    'temio ' = temperature I/O unit (0 no I/O)
   36    'salio ' = salinity    I/O unit (0 no I/O)
    0    'tthio ' = density     I/O unit (0 no I/O)
EOF
}
################################################
E=$Eenv
GLOBAL=$AREA
DOMAINDIRHOME=$ARCH_area
EXPTDIRSCR=$WRK_dir
DOMAIN=hycom
TOPOVER=$Tenv
TOPO=$Topo_area
#
# --- hycom archive output which are input files to this program
#
OUTPUT=${NC_area}
mkdir -p $OUTPUT
INPUT=${DOMAINDIRHOME}
#
# --- output directory for output (netcdf)
#
NCDIR=$OUTPUT
mkdir -p $NCDIR # should already be there
LOGS=$OUTPUT/logs
mkdir -p $LOGS
#
cd $EXPTDIRSCR
#
touch regional.depth.a regional.depth.b
/bin/rm -f regional.depth.[ab]
ln -sf $TOPO/depth_${GLOBAL}_${TOPOVER}.a regional.depth.a
ln -sf $TOPO/depth_${GLOBAL}_${TOPOVER}.b regional.depth.b
#
touch regional.grid.a regional.grid.b
/bin/rm -f regional.grid.[ab]
ln -sf $TOPO/regional.grid.a regional.grid.a
ln -sf $TOPO/regional.grid.b regional.grid.b
#
alias jul2cal=$MODASbin/doy2idtg1
#
# --- loop thru files (1 pero snapshot in julian days)
#
for run in ncdf2d ncdf3d; do
 yyyy=$year
  day=$jul1
```

```
  dend=$jul2
  while [ $day -le  $dend ] ; do

     fday=`echo $day | awk '{printf "%3.3d", $1}'`
     dtg=`jul2cal $day $yyyy`
     echo $dtg
     # The outer nests have the experiment in front of the file
hr="00 06 12 18"
hr="00"
for tau in $hr
do
     HYCOMINPUTFILE=${INPUT}/archv_${GLOBAL}.${yyyy}_${fday}_${tau}.a

     if [ ! -e $HYCOMINPUTFILE ]; then
  echo "file not found $HYCOMINPUTFILE" >$LOGS/${yyyy}_${fday}.err
       # keep processing
     fi
    $run $HYCOMINPUTFILE
done
    day=`echo $day+1 | bc`
  done
 done
```

3.0 RELO NCOM TEST CASES FOR THE NEW JERSEY COAST

This section briefly describes the sensitivity of Relo NCOM to the vertical resolution of the boundary conditions obtained from GOFS V3.0. A regional Relo NCOM is configured for the area off the New Jersey coast bounded by the region 76°-68°W, 37.5°-43.5°N. The horizontal resolution is 3 km in both longitude and latitude and it is configured with 40, 50 and 100 vertical levels (40z, 50z and 100z, respectively). In each configuration, it receives GOFS V3.0 BCs with the same vertical resolution and the 40z case uses the same pre-defined levels noted in section 2.0. It uses wind forcing from the 18 km resolution Coupled Ocean Atmosphere Mesoscale Prediction System (COAMPS) and thermal forcing from the 0.5° Navy Operational Global Atmospheric Prediction System (NOGAPS). It includes river discharge (Barron and Smedstad, 2002) and barotropic tidal forcing from Egbert and Erofeeva (2002) is specified at the open boundaries. The hindcasts span ~2 months (22 May 2009 – 31 July 2009) and observations are assimilated via NCODA *with the exception of profile data*. These are withheld to be used as an

independent validation dataset. GOFS V3.0 BCs are provided to the inner model every 6 hours along the outermost grid row/column. NCODA is turned off within a 10 point buffer zone along each sidewall. The analysis period spans the entire month of June 2009. Figure 1 shows the location of the independent observations.

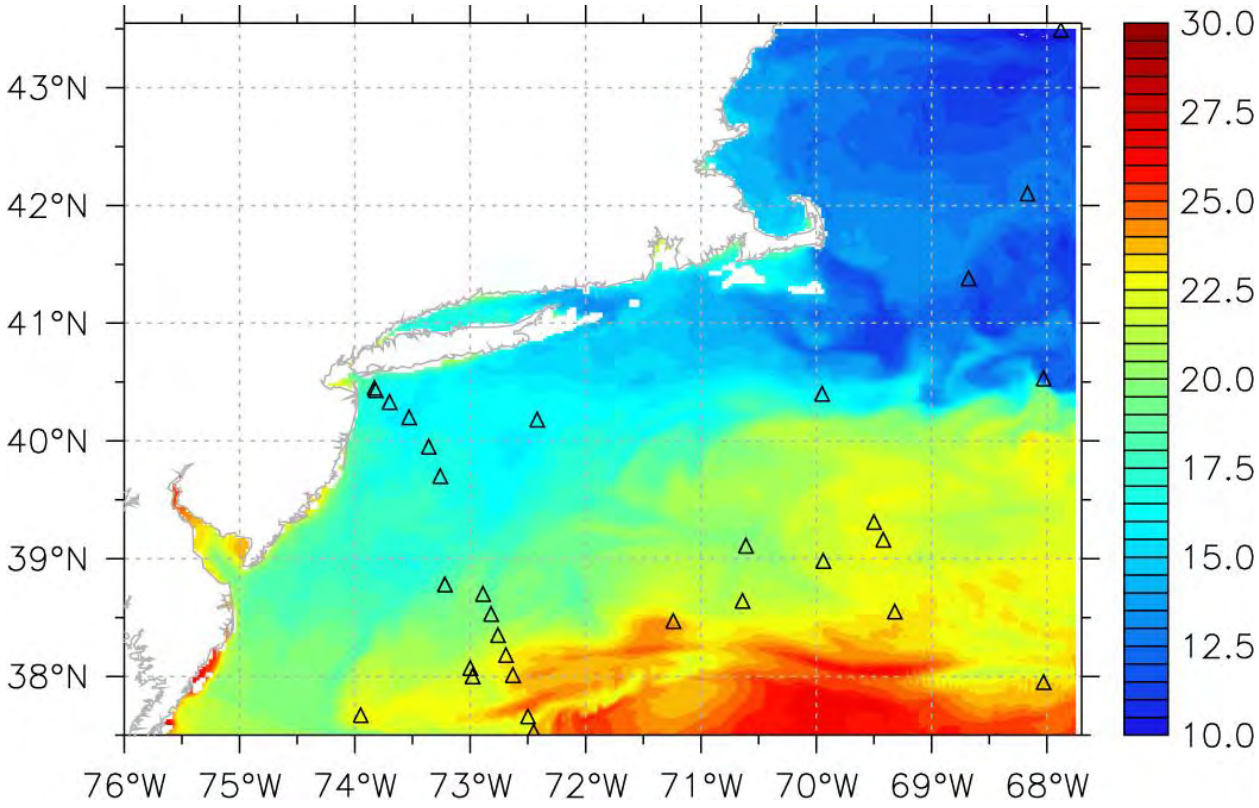

Figure 1: Sea surface temperature (°C) for 15 June 2009 from New Jersey Relo NCOM. The triangles indicate the location of the independent observations. All are Argo profiles with the exception of one moored buoy in the northeast corner of the domain that provided observations most days of the analysis month.

Similar to the GOFS V3.0 Phase II VTR, a temperature versus depth error analysis was performed using independent profiles and is shown in Figure 2. In the upper 200 m of the water column, 50z Relo NCOM has reduced mean error and root mean square error (RMSE) than the 40z and 100 z configurations.

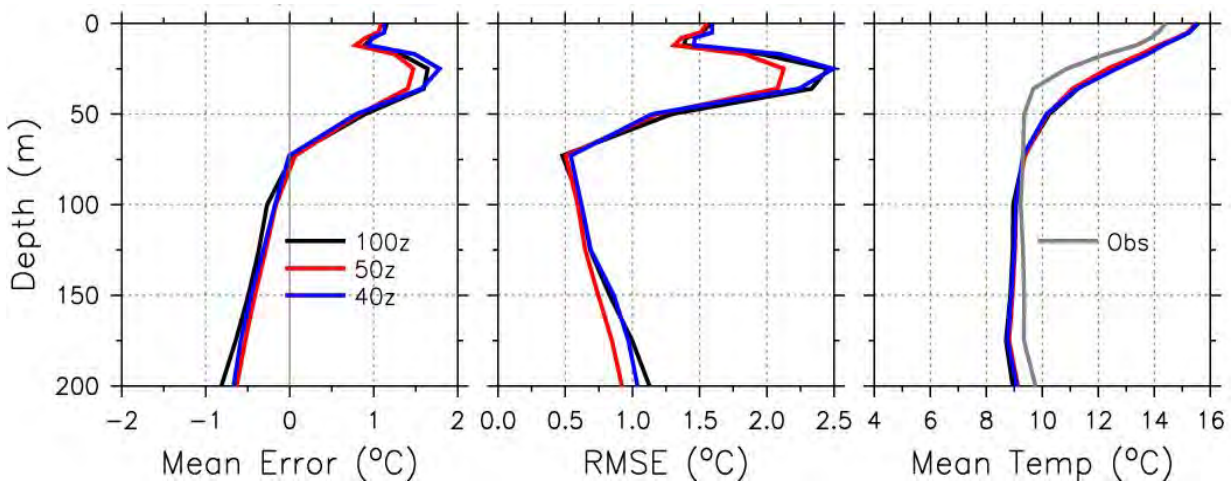

Figure 2: Temperature (°C) versus depth error analysis in the upper 200 m for New Jersey Relo NCOM for June 2009 using 53 unassimilated profiles depicted in Figure 1. The left panel is mean error (bias), the middle panel is RMSE and the right panel an average temperature of all profiles. The gray curve is the observations and Relo NCOM configured with 40 levels (blue), 50 levels (red) and 100 levels (black).

An error analysis of the acoustical proxy variables is also computed and shown in Table 1. The observational profiles did not all contain enough information to compute each variable, and so the number of independent profiles in also listed in column 2. Note, that the analysis for SLD is only based on eight profiles. The mean of each variable is listed in column 3. The region is defined by shallow mixed and sonic layer depths during this time frame. Again, 50z Relo NCOM is the configuration that produces the lowest overall error.

The difference in these results is a consequence of the interpolation methodology employed here. The GOFS V3.0 native grid files are linearly interpolated onto the vertical levels of the netCDF files, and these are further interpolated onto the sigma-z levels of the Relo NCOM domain using the Piecewise Cubic Hermite Interpolating Polynomial (PCHIP). Linear interpolation provides line segments with sharp corners at the data points, while higher order interpolations are made for matching not only the data values but also the slopes and concavities

of each interpolating segment. The PCHIP method modifies the derivative values in the Hermite representation in order to eliminate the bumps and wiggles that are frequently corrupting the fields interpolated by cubic splines or Akiwa algorithms (Fritsch and Carlson, 1980). Therefore, the number of z levels in the netCDF files must be chosen to preserve the accuracy of GOFS V3.0 hybrid vertical coordinates, yet without over constraining the interpolated variables to a high number of linear segments in such a way that the PCHIP algorithm may perform in an area of strong gradients (i.e. the thermocline). In our example the 40 levels are not sufficient to provide a correct parameterization of the GOFS V3.0 thermocline and the 100 levels are over constraining the derivative values for an efficient application of the PCHIP method.

Table 1: Acoustical proxy error analysis for New Jersey Relo NCOM configured for 40, 50 and 100 levels for June 2009 using the number of unassimilated profiles noted in column 2. Those cells highlighted in green have the lowest bias or error of the three configurations.

Variable	# of profiles	Obs. Mean	40z		50z		100z	
			Mean Error	RMSE	Mean Error	RMSE	Mean Error	RMSE
MLD	52	4.9 m	0.0 m	5.6 m	-0.4 m	5.1 m	-0.6 m	5.2 m
SLD	8	30.0 m	6.1 m	9.7 m	-1.9 m	2.4 m	-1.9 m	2.4 m
BLG	49	5.3 m/s/100 ft	0.7 m/s/100 ft	2.3 m/s/100 ft	0.6 m/s/100 ft	2.1 m/s/100 ft	0.8 m/s/100 ft	2.2 m/s/100 ft
DSC	34	278.9 m	-18.9 m	74.8 m	-17.8 m	95.1 m	-26.4 m	88.6 m

4.0 SUMMARY

This report documents the two methodologies for extracting GOFS V3.0 boundary conditions for use in Relo NCOM. In the first method netCDF files from the global system are on a constant .08° latitude/longitude grid and have been vertically remapped to 40 pre-defined levels. The second methodology allows more flexibility it uses the GOFS V3.0 archive files on the native hybrid vertical grid that can be remapped in the vertical to a user defined set of z-levels.

A version of Relo NCOM was configured for the area off the New Jersey coast and configured for 40, 50 and 100 levels, receiving outer BCs from GOFS V3.0 at the same vertical resolution. Using independent profile observations, error analyses of temperature as a function of depth and acoustical proxy variables indicate the configuration with 50 levels had overall slightly lower error than the other configurations.

5.0 ACKNOWLEDGEMENTS

This work was funded as part of the NRL 6.4 Large Scale Prediction project, managed by the Space and Naval Warfare Systems Command under program element 0603207N. The numerical simulations were performed on the Navy DoD Supercomputing Resource Center IBM-Power 6 and Cray XT5 at Stennis Space Center, Mississippi using grants of computer time from the Department of Defense High Performance Computing Modernization Program.

6.0 APPENDIX

The following describes an example of how to process GOFS V3.0 archive files where output exists only over a pre-defined subregion of the global domain. These are on the hybrid vertical grid and can be created as the system runs in real-time. They are designated as the *archt* files and the subregion is relative to the i/j grid as defined in the archt.input file. A sequence of scripts/software must be run to process and make them ready for the next step of extracting BCs for Relo NCOM, i.e. section 2.2.

If the subregion is new, files that define the latitude-longitude grid and topographic depths must be created. This need only be done once. As mentioned above, the subregion is defined in archt.input and may look something like:

```
eslogin1 92> cat archt.input
```

```
0
1
0386 1112 0314 0331
```

where the last line defines the subregion and is listed as: first i-coordinate, last i-coordinate, first j-coordinate, and last j-coordinate. Again, these are relative to the global domain. If the **EXACT** latitude and longitude of the lower left and upper right corners of the subregion are known, then **prep0.make_ij.com** can be used to extract the i,j coordinates.

```
davinci 98% csh prep0.make_ij.com
USAGE   : prep0.make_ij.com -lon1_xx*     -lon2_xx*     -lat1_xx*    -
lat2_xx*
EXAMPLE:  prep0.make_ij.com  -lon1_283.25  -lon2_292.75  -lat1_35.60  -
lat2_42.85

davinci 99% csh prep0.make_ij.com -lon1_105.0 -lon2_130.0 -lat1_-30.0
-lat2_-5.0
i 386 j 1112 idm 314 jdm 331
```

Use **prep1.make_grid.com** to create the subregion file that defines the latitudes and longitudes and **prep2.make_depth.com** to create the subregion file that defines the topographic depths. Note: RRRr0.08 should be replaced with a subregion identifier name and environmental variable $T is the topography version number used in GOFS V3.0.

prep1.make_grid.com

```
#!/bin/csh
#
# Purpose: Create the Sub-Regional & Sub-Depth files.
# Input Files: The Global Regional & Depth files
#
# Executable for Area : sub_grid
# Executable for Depth: isub_topog
#
# Version: 2.0 2011
# By : D.S. Franklin, QinetiQ-NA
#
#################################################################
#
# Gather these variables to the necessary points
#
  set debug = 0
#
  foreach arg ( ${argv} )
    switch ( ${arg} )
      case -lon1_* :
        setenv LON1 `echo ${arg} | sed 's/-lon1_//'`
```

```
        breaksw
      case -lon2_* :
        setenv LON2 `echo ${arg} | sed 's/-lon2_//'`
        breaksw
      case -lat1_* :
        setenv LAT1 `echo ${arg} | sed 's/-lat1_//'`
        breaksw
      case -lat2_* :
        setenv LAT2 `echo ${arg} | sed 's/-lat2_//'`
        breaksw
      case -i_* :
        setenv I1 `echo ${arg} | sed 's/-i_//'`
        breaksw
      case -j_* :
        setenv J1 `echo ${arg} | sed 's/-j_//'`
        breaksw
      case -idm_* :
        setenv IDM `echo ${arg} | sed 's/-idm_//'`
        breaksw
      case -jdm_* :
        setenv JDM `echo ${arg} | sed 's/-jdm_//'`
        breaksw
      case -d :
        set debug = 1
        breaksw
      default:
        echo "USAGE : `basename $0` -lat1_x* -lon1_x* -lat2_x* -
lon2_x*"
        echo "USAGE : or"
        echo "USAGE : `basename $0` -i_* -j_* -idm_* -jdm_*"
        exit 1
        breaksw
    endsw
  end
#
# Global Region is currently 4500 x 3298
#
  if ( ! $?R ) setenv R "GLBa0.08" # Original Global Region
  if ( ! $?S ) setenv S "RRRr0.00" # New SubRegional Name
  if ( ! $?T ) setenv T "09"       # Bathymetry
#
  if ( ! $?SRC ) setenv SRC      "~wallcraf/hycom/ALL/bin"
  if ( ! $?L   ) setenv L  `pwd`
  if ( ! $?T1  ) setenv T1  "${HOME}/hycom/${R}/topo"
  if ( ! $?T2  ) setenv T2  "${HOME}/hycom/${S}/topo"
#
#############################################################################
#
# No need to change anything from below this point
#
#############################################################################
#
# Make sure coordinates exist
#
  if ( ! $?I1 || ! $?J1 || ! $?IDM || ! $?JDM ) then
    if ( $?LAT1 && $?LAT2 && $?LON1 && $?LON2 ) then
      setenv   IJINFO   `/bin/csh   ${L}/make_IJ.com   -lat1_${LAT1}   -
lat2_${LAT2} -lon1_${LON1} -lon2_${LON2}`
```

```
      setenv I1  `echo ${IJINFO} | awk '{ print $2 }'`
      setenv J1  `echo ${IJINFO} | awk '{ print $4 }'`
      setenv IDM `echo ${IJINFO} | awk '{ print $6 }'`
      setenv JDM `echo ${IJINFO} | awk '{ print $8 }'`
    else
      echo "USAGE : `basename $0` -lat1_x* -lon1_x* -lat2_x* -lon2_x*"
      echo "USAGE : or"
      echo "USAGE : `basename $0` -i_* -j_* -idm_* -jdm_*"
      exit 2
    endif
  endif
  echo "CREATING DEPTH FILE for ${S} STARTING AT ${I1}i ${J1}j
COVERING ${IDM}x${JDM} POINTS"
#
  if ( -d ${T1} ) then
    cd ${T1}
  else
    echo "NO Directory found: ${T1}"
    exit 2
  endif
  if ( ! -d ${T2} ) /bin/mkdir -p ${T2}
#
  set grid1 = "${T1}/regional.grid"
  set grid2 = "${T2}/regional.grid"
#
# Remove any previous versions of the subregional files
#
  if ( -e ${grid2}.a ) /bin/rm ${grid2}.a
  if ( -e ${grid2}.b ) /bin/rm ${grid2}.b
#
  echo "MAKING GRID FILE for Region ${S}"
  ${SRC}/../subregion/src/sub_grid <<E-o-D
${grid1}.b
${grid2}.b
${IDM}     'idm   ' = longitudinal array size
${JDM}     'jdm   ' = latitudinal  array size
${I1}      'irefi ' = longitudinal input  reference location
${J1}      'jrefi ' = latitudinal  input  reference location
    1      'irefo ' = longitudinal output reference location
    1      'jrefo ' = latitudinal  output reference location
E-o-D
  if ( -e ${grid2}.a ) then
    echo "CREATED ${grid2}.a"
  else
    echo "UNABLE TO CREATE ${grid2}.a"
    exit 3
  endif
#
# End Program
```

prep2.make_depth.com

```
#!/bin/csh
# SubRegDepth.com
#
# Purpose: Create the Sub-Regional & Sub-Depth files.
# Input Files: The Global Regional & Depth files
#
# Executable for Area : sub_grid
```

```
# Executable for Depth: isub_topog
#
# Version: 2.0 2011
# By : D.S. Franklin, QinetiQ-NA
#
#####################################################################
#
# Gather these variables to the necessary points
#
  set debug = 0
#
  foreach arg ( ${argv} )
    switch ( ${arg} )
      case -lon1_* :
        setenv LON1 `echo ${arg} | sed 's/-lon1_//'`
        breaksw
      case -lon2_* :
        setenv LON2 `echo ${arg} | sed 's/-lon2_//'`
        breaksw
      case -lat1_* :
        setenv LAT1 `echo ${arg} | sed 's/-lat1_//'`
        breaksw
      case -lat2_* :
        setenv LAT2 `echo ${arg} | sed 's/-lat2_//'`
        breaksw
      case -i_* :
        setenv I1 `echo ${arg} | sed 's/-i_//'`
        breaksw
      case -j_* :
        setenv J1 `echo ${arg} | sed 's/-j_//'`
        breaksw
      case -idm_* :
        setenv IDM `echo ${arg} | sed 's/-idm_//'`
        breaksw
      case -jdm_* :
        setenv JDM `echo ${arg} | sed 's/-jdm_//'`
        breaksw
      case -d :
        set debug = 1
        breaksw
      default:
        echo "USAGE : `basename $0` -lat1_x* -lon1_x* -lat2_x* -
lon2_x*"
        echo "USAGE : or"
        echo "USAGE : `basename $0` -i_* -j_* -idm_* -jdm_*"
        exit 1
        breaksw
    endsw
  end
#
# Global Region is currently 4500 x 3298
#
  if ( ! $?R ) setenv R "GLBa0.08" # Original Global Region
  if ( ! $?S ) setenv S "RRRr0.00" # New SubRegional Name
  if ( ! $?T ) setenv T "09"       # Bathymetry
#
  if ( ! $?SRC ) setenv SRC "~wallcraf/hycom/ALL/bin"
  if ( ! $?L   ) setenv L   `pwd`
```

```
  if ( ! $?T1  ) setenv T1  "${HOME}/hycom/${R}/topo"
  if ( ! $?T2  ) setenv T2  "${HOME}/hycom/${S}/topo"
#
#############################################################################
#
# No need to change anything from below this point
#
#############################################################################
#
# Make sure coordinates exist
#
  if ( ! $?I1 || ! $?J1 || ! $?IDM || ! $?JDM ) then
    if ( $?LAT1 && $?LAT2 && $?LON1 && $?LON2 ) then
      setenv   IJINFO   `/bin/csh   ${L}/make_IJ.com   -lat1_${LAT1}   -
lat2_${LAT2} -lon1_${LON1} -lon2_${LON2}`
      setenv I1  `echo ${IJINFO} | awk '{ print $2 }'`
      setenv J1  `echo ${IJINFO} | awk '{ print $4 }'`
      setenv IDM `echo ${IJINFO} | awk '{ print $6 }'`
      setenv JDM `echo ${IJINFO} | awk '{ print $8 }'`
    else
      echo "USAGE : `basename $0` -lat1_x* -lon1_x* -lat2_x* -lon2_x*"
      echo "USAGE : or"
      echo "USAGE : `basename $0` -i_* -j_* -idm_* -jdm_*"
      exit 2
    endif
  endif
  echo  "CREATING  DEPTH  FILE  for  ${S}  STARTING  AT  ${I1}i  ${J1}j
COVERING ${IDM}x${JDM} POINTS"
#
  if ( -d ${T1} ) then
    cd ${T1}
  else
    echo "NO Directory found: ${T1}"
    exit 2
  endif
  if ( ! -d ${T2} ) /bin/mkdir -p ${T2}
#
  set depth1 = "${T1}/depth_${R}_${T}"
  set depth2 = "${T2}/depth_${S}_${T}"
#
# Remove any previous versions of the subregional files
#
  if ( -e ${depth2}.a ) /bin/rm ${depth2}.a
  if ( -e ${depth2}.b ) /bin/rm ${depth2}.b
#
# Create the depth file. This requires a special depth file to take
into
# account the tiled regions. The resulting global file will be large
as all
# the cells are written and filled.
#
  echo "MAKING DEPTH FILE for ${S}"
  ${SRC}/../subregion/src/isub_topog <<E-o-D
${depth1}.b
${depth2}.b
${depth1} subregioned to ${S} via isub_topog
${IDM}     'idm   ' = longitudinal array size
${JDM}     'jdm   ' = latitudinal  array size
```

```
${I1}      'irefi ' = longitudinal input  reference location
${J1}      'jrefi ' = latitudinal  input  reference location
    1      'irefo ' = longitudinal output reference location
    1      'jrefo ' = latitudinal  output reference location
    1       'ijgrd ' = integer scale factor between input and output
grids
E-o-D
  if ( -e ${depth2}.a ) then
    echo "CREATED ${depth2}.a"
  else
    echo "UNABLE TO CREATE ${depth2}.a"
    exit 4
  endif
#
# End Program
```

A global HYCOM bathymetry file specifically for the tiled region is necessary to create the subregion HYCOM archive files. This file should exist as /u/home/${user}/hycom/GLBa0.08/topo/depth_GLBa0.08_tileRRR.[ab]. However this needs a *combined* global HYCOM archive file of the tiled regions that can be created using the script **prep3.make_tile.com**. The IDM and JDM are for the entire global domain. This resultant bathymetry file puts land everywhere the HYCOM file has a p-grid data void. If this is not done, there will be the following error:

```
error - wrong bathymetry for this archive
number of topo sea  mismatches =  <num>
number of topo land mismatches =  <num>
```

prep3.make_tile.com

```
#!/bin/csh
#
# Program: make_tile.com
# Create a masked tiled Global Depth file
# --- bathyetry for expt 74.2 subregion archive.
# Since directory structure changes for the archt files,
# pass directory information via command line
#
# Version: 2.0 2011
# By : D.S. Franklin, QinetiQ-NA
# Modified Script from Alan Wallcraft (NRL)
#
################################################
#
setenv R "GLBa0.08"
setenv V "RRRr0.08"
setenv U `echo ${V} | cut -c1-3`
setenv E "185"
```

```
setenv X `echo ${E}  | awk '{ printf("%04.1f\n", $1 * 0.1 ) }'`
setenv T "09"
setenv Y03  108
setenv YY `echo ${Y03} | awk '{ print $1 + 1900 }'`
setenv A  "h"
setenv O "${YY}_244"
setenv G "01"
#
setenv SRC  "~wallcraf/hycom/ALL/bin"
setenv T1  "${HOME}/hycom/${R}/topo"
setenv T2  "${HOME}/hycom/${V}/topo"
setenv TMPF "tmp.lis"
setenv OD  "${T1}/depth_${R}_${T}"
setenv GD  "depth_${R}_${T}_tile${U}"
setenv S   "/scr/${user}/hycom/${R}/expt_${X}/data/tart_${Y03}${A}"
setenv L   "${HOME}/hycom/${V}/subregion"
setenv GF  "${S}/${E}_archt.${O}_${G}"
#
  cd ${S}
#
# Check for existance of a a full days worth of global archive data
# Create if missing.
#
  if ( ! -e ${GF}.a ) then
#
#   Link topo files for global processing
#
    if ( -e ./regional.grid.a ) /bin/rm ./regional.grid.a
    if ( -e ./regional.grid.b ) /bin/rm ./regional.grid.b
    /bin/cp  ${HOME}/hycom/${R}/topo/regional.grid.a .
    /bin/cp  ${HOME}/hycom/${R}/topo/regional.grid.b .
#
#   Create a a full days worth of global archive data
#
    awk -f ${L}/subreg.awk o=${O} g=${G} r=${R} e=${E} x=${X} t=${T}
s="${S}" \
          ${L}/step1.make_global.com  >!
${E}_make_global_${O}_${G}.com
    csh ${E}_make_global_${O}_${G}.com >&!
${E}_make_global_${O}_${G}.log
  endif
#
# --- bathyetry for expt ${X} subregion archive.
#
  if ( -e ${GF}.a ) then
    if ( -e ${GD}.a ) /bin/mv ${GD}.a++
    if ( -e ${GD}.b ) /bin/mv ${GD}.b++

    setenv IDM `grep idm ${T1}/regional.grid.b | awk '{ print $1 }'`
    setenv JDM `grep jdm ${T1}/regional.grid.b | awk '{ print $1 }'`
#
    if ( -e ${TMPF}      ) /bin/rm -f ${TMPF}
    ${SRC}/hycom_mask ${OD}.a ${GF}.a ${IDM} ${JDM} ${GD}.a > ${TMPF}
    if ( ! ${status} ) then
      echo "Failed to create ${GD}.a, exiting"
      exit 5
    endif
#
```

```
# --- uses min,max from hycom_mask
#
   set min = `grep min ${TMPF} | awk '{ print $4 }'`
   set max = `grep max ${TMPF} | awk '{ print $5 }'`
cat <<E-o-D >! ${GD}.b
from depth_${R}_${T}
MASKED EVERYWHERE EXCEPT FOR EXPT ${X} SUBREGION ARCHIVE LOCATIONS
based on smith and sandwell 1min

Depths greater than  6500.0 scaled by 0.20
E-o-D
printf "min,max depth = %10.3f %9.3f\n" $min $max >> ${GD}.b
#
    cat ${GD}.b
    /bin/mv -f ${GD}.[ab] ${T1}/
  else
    echo "ERROR : Missing Tiled Global Archive File ${GF}"
  endif
#
# End Program
```

In the main subregional topo directory a new file of the type "**RegionIJ.csh**" is required to have the IDM, JDM, IREF, JREF and file size information in the C-Shell environmental setup for future sourcing. This is created by running **prep4.make_RegionIJ.com**.

prep4.make_RegionIJ.com

```
#!/bin/csh
setenv R "GLBa0.08"
setenv V "RRRr0.08"
setenv W "~wallcraf/hycom/ALL/bin/"
setenv I "Region_${V}_IJ.csh"
#
# Determine if the file already exists,
# if so back it up into a unique name before recreating.
#
if ( -e ${HOME}/hycom/${V}/topo/${I} ) then
  /bin/mv                -f             ${HOME}/hycom/${V}/topo/${I}
${HOME}/hycom/${V}/topo/${I}.`date +"%Y%m%d%H%M"`
endif
#
# From the Subregion regional grid file, find the IJ domain
#
setenv IDM `grep idm ${HOME}/hycom/${V}/topo/regional.grid.b | awk '{
print $1 }'`
setenv JDM `grep jdm ${HOME}/hycom/${V}/topo/regional.grid.b | awk '{
print $1 }'`
#
# Use the Subregion regional grid file to determine
# the start of the region in global lonlat coordinates.
#  Use  this  lonlat  to  determine  the  reference  point  on  the  global
regional grid.
#
```

```
set        ll        =        `${W}/hycom_ij2lonlat        1        1
${HOME}/hycom/${V}/topo/regional.grid.a`
set ll = `echo ${ll} | sed -e 's/E//' -e 's/N//' -e 's/\(.*\)W\(.*\)/-
\1\2/' -e 's/\(.*\) \(.*\)S\(.*
\)/\1 -\2/'`
set          ll          =          (`${W}/hycom_lonlat2ij          ${ll}
${HOME}/hycom/${R}/topo/regional.grid.a`)
#
# Create the regional IJ file
#
echo '  setenv IDM     "'${IDM}'"' >! ${HOME}/hycom/${V}/topo/${I}
echo '  setenv JDM     "'${JDM}'"' >> ${HOME}/hycom/${V}/topo/${I}
echo '  setenv IREFI   "'$ll[1]'"' >> ${HOME}/hycom/${V}/topo/${I}
echo '  setenv JREFI   "'$ll[2]'"' >> ${HOME}/hycom/${V}/topo/${I}
```

Creating the tiled HYCOM archive file uses the steps in the script **step0.submitwrap.com**. Modify this with the experiment number, the subregions, etc. This queue submission wrapper script creates a global HYCOM archive file per hour (**step1.make_global.com**), each regional subfile per hour (**step2.isubreghrly.com**), and finally will tar up the resultant subregional hourly files into daily archives (**step3.tarhrfiles.com**).

All scripts are created using the HYCOM technique of an awk script, "subreg.awk", to modify the resultant scripts as needed. The user should only have to modify **step0.submitwrap.com**, and no changes should be required in **step1.make_global.com**, **step2.isubreghrly.com**, and **step3.tarhrfiles.com**.

step0.submitwrap.com

```
#!/bin/csh
#PBS -N XXX
#PBS -j oe
#PBS -o XXX.log
#PBS -l walltime=6:00:00
#PBS -W umask=027
#PBS -A ACCTNUM
#PBS -q standard
#
# Wrapper script to run all the processing
#
# Version: 2.0 2011
# By : D.S. Franklin, QinetiQ-NA
#      Based on Scripts by E.J.Metzger, NRL
```

```
#
set echo
set verbose
#
setenv OS `uname`
switch ($OS)
case 'SunOS':
case 'Linux':
    which yod
    if (! $status) then
      setenv OS XT3
    endif
    which aprun
    if (! $status) then
C --- XT4 or XT5
      setenv OS XT4
      set APRUN='aprun -n 1'
      set APRUNPLT='aprun -n 1 '
    endif
    breaksw
case 'AIX':
     set APRUN=''
     set APRUNPLT=''
    breaksw
default:
set APRUN=''
 set APRUNPLT=''
endsw
#
# --- convert tiled to standard archives
#
# Set the following switches
#  -- 1 processing
#  -- 0 skip
#  -- global - create hourly global archive file
#  -- isub   - create hourly subregional file
#  -- tarNrm - tar up subregional files and remove global and
subregional files
#
set global = 1
set isub   = 1
set tarNrm = 0
#
setenv R GLBa0.08
setenv E 185
setenv X 18.5
setenv T 09
setenv VS "RRRr0.08"
setenv I  "Region_${VS}_IJ.csh"
#
# - environment variables defining the year and part
#
setenv Y03  108
setenv YY `echo ${Y03} | awk '{ print $1 + 1900 }'`
setenv A    h
setenv GLB_HOURS "00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16
17 18 19 20 21 22 23"
setenv DTG1 214
```

```
setenv DTG2 244
setenv Z  "1000"
#
# - environment variables defining the working directories
#
setenv S  /scr/${user}/hycom/${R}/expt_${X}/data/tart_${Y03}${A}
setenv L  ${HOME}/hyc2ncom/subregion
#
date +"STATUS : %D %T : Using arguments :"
date +"STATUS : %D %T :                    REGIONS   = ${VS}"
date +"STATUS : %D %T :                    GLB_HOURS = ${GLB_HOURS}"
if ( ! ${global} ) date +"WARNING: %D %T : SKIPPING GLOBAL Processing"
if ( ! ${isub}   ) date +"WARNING: %D %T : SKIPPING REGIONAL
Processing unless missing"
#
while ( `echo ${DTG1} | sed 's/^0*//'` <= `echo ${DTG2} | sed
's/^0*//'` )

  if ( `echo ${DTG1} | wc -c` > 8  ) then
#
#   If Ordinal day is from DTG1: Output is in form YEAR_ORD
#
    setenv Y    `echo ${DTG1} | cut -c1-4`
    setenv MTH  `echo ${DTG1} | cut -c5-6`
    setenv DAY  `echo ${DTG1} | cut -c7-8`
    setenv O    `${L}/ymd2doy.csh ${Y} ${MTH} ${DAY}`
  else
    setenv O ${YY}_${DTG1}
  endif
  date +"STATUS : %D %T : Processing for ${O}"
#
# GLOBAL PROCESSING
#
  if ( ${global} ) then

    cd ${S}
#
#   Link topo files for global processing
#
    if ( -e ./regional.grid.a ) /bin/rm ./regional.grid.a
    if ( -e ./regional.grid.b ) /bin/rm ./regional.grid.b
    /bin/cp  ${HOME}/hycom/${R}/topo/regional.grid.a .
    /bin/cp  ${HOME}/hycom/${R}/topo/regional.grid.b .
#
#   Create a full days worth of global data
#
    foreach G ( ${GLB_HOURS} )
      awk -f ${L}/subreg.awk o=${O} g=${G} r=${R} e=${E} x=${X} t=${T}
z=${Z} s="${S}" \
             ${L}/step1.make_global.com  >!
${E}_make_global_${O}_${G}.com
      csh ${E}_make_global_${O}_${G}.com >&!
${E}_make_global_${O}_${G}.log &
#
#     Set automatically the size of the file for the required region
#
      if ( -e ${E}_archt.${O}_${G}.a ) then
        set sz = `ls -l ${E}_archt.${O}_${G}.a | awk '{ print $5 }'`
```

```
        if ( "${sz}" > "${Z}" ) setenv Z ${sz}
      endif
    end
    wait
#
#   Sometimes the global file is not created, rerun if missing
#
    foreach G ( ${GLB_HOURS} )
      if ( ! -e ${E}_archt.${O}_${G}.a ) then
        date +"WARNING: %D %T : First pass did not create
${E}_archt.${O}_${G}.a, trying again"
        awk -f ${L}/subreg.awk o=${O} g=${G} r=${R} e=${E} x=${X}
t=${T} z=${Z}  s="${S}" \
        ${L}/step1.make_global.com  >!  ${E}_make_global_${O}_${G}.com
        csh ${E}_make_global_${O}_${G}.com >&!
${E}_make_global_${O}_${G}.log
        if ( ! -e ${E}_archt.${O}_${G}.a ) then
          date +"ERROR  : %D %T : Unable to create
${E}_archt.${O}_${G}.a"
        endif
      endif
    end
    echo "FINISHED Creating Global Data file"
  endif
#
# REGIONAL PROCESSING
#
  if ( ${isub} ) then
    foreach V ( ${VS} )
      setenv U `echo ${V} | cut -c1-3`
      cd ${S}
      if ( -e ${HOME}/hycom/${V}/topo/${I} ) then
#
#       Source specific dimensions of region
#
        /bin/cp ${HOME}/hycom/${V}/topo/${I} .
        source  ${HOME}/hycom/${V}/topo/${I}
#
#       Create regional archive file
#
        foreach G ( ${GLB_HOURS} )
#
#         Check that a tile file has been created
#
          set tfile = ${HOME}/hycom/${R}/topo/depth_${R}_${T}_tile${U}
          if ( ! -e ${tfile}.a ) then
            awk -f ${L}/subreg.awk o=${O} g=${G} r=${R} e=${E} x=${X}
t=${T} \
                     v=${V} idm=${IDM} jdm=${JDM} ir=${IREFI}
jr=${JREFI} s="${S}" \
                    ${L}/prep3.make_tile.com >! ${E}_make_tile_${V}.com
            /bin/csh ${E}_make_tile_${V}.com >&!
${E}_make_tile_${V}.log
          endif
          /bin/cp -p ${tfile}.[ab] .
          if ( -e ${E}_archt.${O}_${G}.a ) then
            awk -f ${L}/subreg.awk o=${O} g=${G} r=${R} e=${E} x=${X}
t=${T} \
```

```
                    v=${V} idm=${IDM} jdm=${JDM} ir=${IREFI}
jr=${JREFI} s="${S}" \
                  ${L}/step2.isubreghrly.com >!
${E}_isubreghrly_${O}-${G}_${V}.com
           /bin/csh ${E}_isubreghrly_${O}-${G}_${V}.com >&!
${E}_isubreghrly_${O}-${G}_${V}.log &
         else
           date +"ERROR  : %D %T : Global file ${E}_archt.${O}_${G}.a
missing, can not create subregion"
         endif
       end
       wait
     else
       date +"ERROR  : %D %T : Missing file
${HOME}/hycom/${V}/topo/${I}"
       date +"ERROR  : %D %T : Creating the file necessary to have
the IDMxJDM and length of region, Rerun"
       awk -f ${L}/subreg.awk r=${R} v=${V}
${L}/prep4.make_RegionIJ.com >! ${E}_make_RegionIJ.com
       /bin/csh ${E}_make_RegionIJ.com >&! ${E}_make_RegionIJ.log
     endif
   end
 endif
#
#   At end of the day, after the last hourly file:
#
# Remove hourly files if directed
#
 if ( ${tarNrm} ) then
   foreach V ( ${VS} )
     awk -f ${L}/subreg.awk o=${O} r=${R} e=${E} x=${X} t=${T} \
                              v=${V} gh="${GLB_HOURS}" s="${S}" \
             ${L}/step3.tarhrfiles.com >!
${E}_tarhrfiles_${O}_${V}.com
           /bin/csh ${E}_tarhrfiles_${O}_${V}.com >&!
${E}_tarhrfiles_${O}_${V}.log &
   end
 endif
#
 date +"STATUS : %D %T : End of Processing for ${O}"
 date +"STATUS : %D %T : ====================================="
#
# Update the day by your favorite program
#
 if ( `echo ${DTG1} | sed 's/^0*//'` > 19000000 ) then
   perl ${L}/dtgadj.pl ${DTG1} +d1
 else
   setenv DTG1 `expr ${DTG1} + 1`
 endif
 date +"STATUS : %D %T : New date is ${DTG1}"
end
wait
```

step1.make_global.com

```
#!/bin/csh
#
# Version: 2.0 2011
# By : D.S. Franklin, QinetiQ-NA
```

```
#      Based on Scripts by E.J.Metzger, NRL
#
set echo
set verbose
set gstatus = 0
set gsize   = 0
#
# --- convert tiled to standard global archive file
#
if ( ! $?APRUN ) setenv APRUN ""
setenv R GLBa0.08
setenv E 742
setenv X `echo ${E} | awk '{ printf("%04.1f\n", $1*0.1 ) }'`
setenv T 09
setenv O 2007_001
setenv G 00
#
# - environment variables defining the working directories
# - tart_2007083118_2007090100
#
setenv S "/scr/${user}/hycom/${R}/expt_${X}/data"
setenv Z "10000000000"
setenv SRC "~wallcraf/hycom/ALL/bin"
#
date +"STATUS : %D %T : Using arguments :"
date +"STATUS : %D %T :       YEAR_ORD  = ${O}"
date +"STATUS : %D %T :       HOUR      = ${G}"
date +"STATUS : %D %T :   Min File Size = ${Z}"
#
#############################################################
#
# Create GLOBAL file from the tiled files
#
#############################################################
#
  if ( ! -d ${S} ) mkdir -p ${S}
#
# Run over the number of hours in a day
#
  cd ${S}
  date +"STATUS : %D %T : Processing Day ${O} for Hour ${G}"
  if ( ! -e archt2archv ) /bin/cp ${SRC}/../archive/src/archt2archv .
#
# Link topo files for global processing
#
  if ( ! -e regional.grid.a ) /bin/cp
${HOME}/hycom/${R}/topo/regional.grid.a
  if ( ! -e regional.grid.b ) /bin/cp
${HOME}/hycom/${R}/topo/regional.grid.b
#
  set gfile  = "${E}_archt.${O}_${G}"
#
# Remove previous versions of the file
#
  /bin/rm -f  ./${gfile}.[ab] >&! /dev/null
  echo       "./${gfile}.a"   >! ${gfile}.tmp
#
  date +"STATUS : %D %T : Searching for ./?????/archt.${O}_${G}.A"
```

```
  /bin/ls -1 ./?????/archt.${O}_${G}.A  >> ${gfile}.tmp
  set files = `wc -l ${gfile}.tmp | awk '{ print $1 }'`
#
  date +"STATUS : %D %T : Running archt2archv with ${files} files for
${O} ${G} with ${gfile}.tmp"
  ${APRUN} ./archt2archv < ${gfile}.tmp
  set gstatus = ${status}
  set gsize   = 0
  if ( -e ${gfile}.a ) set gsize   = `ls -l ./${gfile}.a | awk '{
print $5 }'`
  if ( ${gsize} < ${Z} ) then
    if ( -e ${gfile}.a ) /bin/rm -f ${gfile}.[ab]
    set gstatus = 5
  endif
  if ( ${gstatus} != 0 ) date +"ERROR  : %D %T : Problem creating
global files for ${O} ${G}"
```

step2.isubreghrly.com

```
#!/bin/csh
#
# Version: 2.0 2011
# By : D.S. Franklin, QinetiQ-NA
#      Based on Scripts by E.J.Metzger, NRL
#
set echo
set verbose
#
# --- convert tiled to standard archive file
#
if ( ! $?APRUN ) setenv APRUN ""
setenv R GLBa0.08
setenv E 742
setenv X `echo ${E} | awk '{ printf("%04.1f\n", $1*0.1 ) }'`
setenv T 09
setenv O 2007_001
setenv G 00
#
#   Set regional variables
#
setenv V   "ITFt0.08"
setenv U   `echo ${V} | cut -c1-3`
setenv IDM "XXX"
setenv JDM "XXX"
setenv IR  "XXX"
setenv JR  "XXXX"
#
# - environment variables defining the working directories
#
setenv S "/scr/${user}/hycom/${R}/expt_${X}/data"
setenv W "~wallcraf/hycom/ALL/bin"
#
setenv T1 "${HOME}/hycom/${R}/topo"
setenv T2 "${HOME}/hycom/${V}/topo"
#
  date +"STATUS : %D %T : Using arguments :"
  date +"STATUS : %D %T : YEAR_ORD  = ${O}"
  date +"STATUS : %D %T : REGION    = ${V}"
#
# Create regional hourly files, this needs a specific tiled depth file created
for it
#
  date +"STATUS : %D %T : Creating regional files for ${O} ${G}"
```

```
#
# Change to regional directory, link needed files, if necessary
#
  if ( ! -d ${S}/${V} ) mkdir -p ${S}/${V}
  cd ${S}/${V}
  if ( ! -e isubregion      ) /bin/cp ${W}/../subregion/src/isubregion .
  if ( ! -e regional.grid.a ) /bin/cp ${T1}/regional.grid.a .
  if ( ! -e regional.grid.b ) /bin/cp ${T1}/regional.grid.b .
#
  set GD = "depth_${R}_${T}_tile${U}"
  if ( ! -e ${GD}.a ) /bin/cp ${T1}/${GD}.a .
  if ( ! -e ${GD}.b ) /bin/cp ${T1}/${GD}.b .
#
  set RD = "depth_${V}_${T}"
  if ( ! -e ${RD}.a ) /bin/cp ${T2}/${RD}.a .
  if ( ! -e ${RD}.b ) /bin/cp ${T2}/${RD}.b .
#
# Set input and output name, removing any previous versions of the output
#
  set input_name  = "${S}/${E}_archt.${O}_${G}"
  set output_name = "${E}_archt_${V}.${O}_${G}"
  /bin/rm ${output_name}.[ab] >&! /dev/null
#
# Create the subregion hourly file
#
  date +"STATUS : %D %T : Running isubregion for ${V} ${O} ${G}"
  ${APRUN} ./isubregion <<E-o-D
${input_name}.b
${GD}.b
${output_name}.b
${RD}.b
${R} interpolated to ${V}
${IDM}    'idm   ' = longitudinal array size
${JDM}    'jdm   ' = latitudinal  array size
${IR}  'irefi ' = longitudinal input  reference location
${JR}  'jrefi ' = latitudinal  input  reference location
   1      'irefo ' = longitudinal output reference location
   1      'jrefo ' = latitudinal  output reference location
   1      'ijgrd ' = integer scale factor between input and output grids
   0      'iceflg' = ice in output archive flag (0=none,1=energy loan model)
   0      'smooth' = smooth interface depths    (0=F,1=T)
E-o-D
```

step3.tarhrfiles.com

```
#!/bin/csh
#
#   At end of the day, after the last hourly file:
#      -  Collect all regions into tar file
#      -  Clean up hourly files
#
# Version: 2.0 2011
# By : D.S. Franklin, QinetiQ-NA
#      Based on Scripts by E.J.Metzger, NRL
#
set echo
set verbose
#
setenv R GLBa0.08
setenv V ITFt0.08
setenv E 742
setenv X 74.2
```

```
setenv GH "00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19
20 21 22 23"
setenv HR ""
setenv O 2007_001
set remove  = 0
set gstatus = 0
#
# - environment variables defining the working directories
#
setenv S /scr/${user}/hycom/${R}/expt_${X}/data
setenv OUT_R  "${S}/${V}"
#
date +"STATUS : %D %T : Using arguments :"
date +"STATUS : %D %T : OUT_R = ${OUT_R}"
date +"STATUS : %D %T : YEAR_ORD  = ${O}"
#
# Create Tar file of all hours in a day
#
cd ${OUT_R}
#
set tarfile = "${E}_archt_${V}.${O}.tar"
foreach h ( ${GH} )
  if ( ! -e "${E}_archt_${V}.${O}_${h}.a" ) then
    date +"WARNING: %D %T : Missing hourly file
${E}_archt_${V}.${O}_${h}.a"
    setenv HR "${HR} ${h}"
    set gstatus = 115
  else
    if ( -e ${S}/${E}_archt.${O}_${h}.a ) then
      date +"STATUS : %D %T : Removing Global file
${E}_archt.${O}_${h}.a"
      if ( -e ${S}/${E}_archt.${O}_${h}.a   ) /bin/rm -f
${S}/${E}_archt.${O}_${h}.a
      if ( -e ${S}/${E}_archt.${O}_${h}.b   ) /bin/rm -f
${S}/${E}_archt.${O}_${h}.b
      if ( -e ${S}/${E}_archt.${O}_${h}.tmp ) /bin/rm -f
${S}/${E}_archt.${O}_${h}.tmp
    endif
  endif
end
if ( ! ${gstatus} || ${gstatus} == "115" ) then
  date +"STATUS : %D %T : Creating tar file ${tarfile}"
  tar -cvf ${tarfile} ${E}_archt_${V}.${O}_??.[ab]
  if ( ${gstatus} != "115" ) set gstatus = ${status}
#
# Remove hourly files after creation of the tar file
#
  date +"STATUS : %D %T : Completed Gtar for tarfile ${tarfile} DTG
${O} with status = ${gstatus}"
  if ( ${gstatus} == 0 && -e ${tarfile} ) then
    ls -l ${tarfile}
    if ( ${remove} ) then
      date +"STATUS : %D %T : Removing hourly Regional Files for ${V}
${O}:"
      /bin/rm -f ${OUT_R}/${E}_archt_${V}.${O}_??.[ab]
    endif
  else if ( ${gstatus} == "115" ) then
    date +"WARNING: %D %T : Missing hourly files : ${HR}"
```

```
  else
    date +"WARNING: %D %T : Unable to create tar file ${tarfile}"
  endif
else
  date +"WARNING: %D %T : Unable to create tar file ${tarfile}"
endif
```

7.0 REFERENCES

Barron, C. N. and L.F. Smedstad, 2002: Global river inflow within the Navy Coastal Ocean Model. *Proc. Oceans 2002 MTS/IEEE Conference*, Biloxi, Mississippi, USA, pp. 1472–1479.

Egbert, G. and S. Erofeeva, 2002: Efficient inverse modeling of barotropic ocean tides. *J. Atmos. Ocean Technol*., **19**, 183-204.

Fritsch, F.N. and R.E. Carlson; 1980: Monotone piecewise cubic interpolation. *SIAM J. Numerical Analysis,* **17,** 238-46.

Metzger, E.J., O.M. Smedstad, P. Thoppil, H.E. Hurlburt, A.J. Wallcraft, D.S. Franklin, J.F. Shriver and L.F. Smedstad, 2008: Validation Test Report for the Global Ocean Prediction System V3.0 - 1/12º HYCOM/NCODA: Phase I. *NRL Memo. Report*, NRL/MR/7320--08-9148.

Metzger, E.J., O.M. Smedstad, P. Thoppil, H.E. Hurlburt, D.S. Franklin, G. Peggion, J.F. Shriver T.L. Townsend and A.J. Wallcraft, 2010: Validation Test Report for the Global Ocean Forecast System V3.0 - 1/12º HYCOM/NCODA: Phase II. *NRL Memo. Report*, NRL/MR/7320--10-9236.

Rowley, C., P.J. Martin and J.A. Cummings, 2010: The Naval Research Laboratory Relocatable Ocean Nowcast/Forecast System. *U.S. Navy Journal of Underwater Acoustics.* **60(1)**, 169-202.

8.0 TABLE OF ACRONYMS

BCs	Boundary Conditions
BLG	Below Layer Gradient
DSC	Deep Sound Channel
DSRC	DoD Supercomputing Research Center
GOFS	Global Ocean Forecast System
HYCOM	HYbrid Coordinate Ocean Model
MLD	Mixed Layer Depth
MODAS	Modular Ocean Data Analysis System
NAVOCEANO	Naval Oceanographic Office
NCODA	Navy Coupled Ocean Data Assimilation
NCOM	Navy Coastal Ocean Model
NLOM	NRL Layered Ocean Model
NRL	Naval Research Laboratory
SLD	Sonic Layer Depth
T	Temperature
VTR	Validation Test Report